

---

# BACHELORARBEIT

---

Herr  
Sebastian Musch

**Konzeption und  
Implementierung eines  
Touchscreenmoduls als  
Kommunikationsplattform der  
Abteilung Informations-  
technologie der Lufthansa  
Technik Logistik GmbH**

Mittweida, 2011

# **BACHELORARBEIT**

---

## **Konzeption und Implementierung eines Touchscreenmoduls als Kommunikationsplattform der Abteilung Informations- technologie der Lufthansa Technik Logistik GmbH**

Autor:  
**Herr  
Sebastian Musch**

Studiengang:  
**Wirtschaftsinformatik**

Seminargruppe:  
**WF07w1-B**

Erstprüfer:  
**Prof. Dr.-Ing. Wilfried Schubert**

Zweitprüfer:  
**Dr.-Ing. Stephanie Nagel-Heyer**

Einreichung:  
**Mittweida, 11.02.2011**

Verteidigung/Bewertung:  
**Mittweida, 2011**

## **Bibliografische Angaben:**

Sebastian, Musch:

Konzeption und Implementierung eines Touchscreenmoduls als Kommunikationsplattform der Abteilung Informationstechnologie der Lufthansa Technik Logistik GmbH – 2011 – 7 S. Verzeichnisse, 68 S. Inhalt, 16 S. Anhang. Mittweida, Hochschule Mittweida (FH), University of Applied Sciences, Fakultät für Mathematik/Naturwissenschaften/Informatik, Bachelorarbeit, 2011

## **Referat:**

Ziel dieser Arbeit ist die Entwicklung eines IT-Konzepts zur internen Kommunikation bei der Lufthansa Technik Logistik GmbH. In diesem Zusammenhang wird ein geeignetes Verfahren zur effektiven und nachhaltigen Datenspeicherung der Inhalte beschrieben. Dazu wird auf allgemeine Content-Management-Systeme und den damit verbundenen Eigenschaften eingegangen. Anschließend wird ein Softwaresystem zur passenden Darstellung der Inhalte entworfen und die Anbindung an ein Content-Management-System erläutert. Eine prototypische Umsetzung der Machbarkeit des Konzepts wird abschließend durch ausgewählte Programmfunktionen und Methoden erläutert.

# Inhalt

<b>Abbildungsverzeichnis .....</b>	<b>VI</b>
<b>Abkürzungsverzeichnis .....</b>	<b>VIII</b>
<b>1. Einleitung.....</b>	<b>9</b>
<b>2. Einführende Grundlagen .....</b>	<b>11</b>
2.1 <i>Lufthansa Technik Logistik GmbH.....</i>	11
2.2 <i>Abteilung Informationstechnologie .....</i>	11
2.2.1    Informationslogistik (ILOG).....	12
2.2.2    Physische Logistik (PLOG) .....	12
2.2.3    Steuerungslogistik (SLOG).....	12
2.3 <i>Interne Kommunikation .....</i>	13
2.3.1    Notwendigkeit der internen Kommunikation .....	13
2.3.2    Arten der unternehmensinternen Kommunikation.....	14
2.3.2.1    Intranet.....	14
2.3.2.2    Mitarbeiterzeitschrift.....	15
2.4 <i>Aufgabenstellung .....</i>	16
2.4.1    Motivation .....	16
2.4.2    Zielstellung.....	16
2.4.3    Abgrenzung.....	17
<b>3. Softwareauswahl.....</b>	<b>18</b>
3.1 <i>Wege der Softwareeinführung und -entwicklung .....</i>	18
3.1.1    Einführung von Standardsoftware .....	19
3.1.2    Eigenentwicklung des Softwareprodukts .....	21
3.2 <i>Auswahl der Umsetzungsvariante .....</i>	23
<b>4. Content Management.....</b>	<b>26</b>
4.1 <i>Content-Management-System.....</i>	26
4.1.1    Strukturierung und Darstellung der Inhalte .....	27
4.1.2    Nutzenpotentiale .....	28
4.2 <i>Enterprise-Content-Management – Alfresco Lösung.....</i>	29
4.2.1    Analyse der bestehenden Informationen .....	31



4.2.2	Content Modeling .....	32
4.2.3	Content Management Interoperability Services .....	34
<b>5.</b>	<b>Systementwurf .....</b>	<b>36</b>
5.1	<i>Anforderungsmanagement</i> .....	36
5.1.1	Funktionelle Anforderungen .....	37
5.1.2	Technische Anforderungen .....	38
5.2	<i>Systemmodellierung</i> .....	39
5.2.1	Produkteinsatz .....	40
5.2.2	Systembeschreibung .....	41
5.2.2.1	Use Cases Alfresco .....	43
5.2.2.2	Use Cases L/information .....	45
5.2.3	Grobentwurf Systemarchitektur .....	46
<b>6.</b>	<b>Entwicklung des Prototyps .....</b>	<b>48</b>
6.1	<i>Entwicklung der Content Struktur mithilfe von Alfresco</i> .....	49
6.1.1	Kategorie Model .....	49
6.1.2	Application Model .....	52
6.1.3	News Model .....	54
6.1.4	Strukturkonzept .....	54
6.2	<i>Monitorapplikation L/information</i> .....	56
6.2.1	Schnittstelle zu Alfresco – Klasse Connector .....	57
6.2.1.1	Verbindungsaufbau zum Content Repository .....	57
6.2.1.2	Abruf der Kategorien .....	58
6.2.1.3	Abruf der Applikationen .....	59
6.2.2	Aufbau und Darstellung der Inhalte .....	61
6.2.2.1	Fensteraufbau .....	61
6.2.2.2	Darstellung der Kategorien .....	63
6.2.2.3	Darstellung der Applikationen .....	64
6.3	<i>Weiterentwicklung und Ausblick</i> .....	66
<b>7.</b>	<b>Zusammenfassung .....</b>	<b>67</b>
	<b>Anhang .....</b>	<b>LXVIII</b>
	<b>Literatur .....</b>	<b>LXXXIV</b>
	<b>Eidesstattliche Erklärung .....</b>	<b>LXXXVI</b>

# Abbildungsverzeichnis

Abbildung 1 – Vorgehensmodell.....	19
Abbildung 2 – Vor- und Nachteile von Standardsoftware.....	21
Abbildung 3 – Vor- und Nachteile von Individualsoftware .....	22
Abbildung 4 – Eigenentwicklung vs. Standardsoftware.....	24
Abbildung 5 – Komponenten des ECM.....	30
Abbildung 6 – Präsentationsfolie; „Blick über den Tellerrand“.....	31
Abbildung 7 – Content Metamodel .....	33
Abbildung 8 – Abbildung von Anwendungsbereichen auf Basiskonzepte .....	40
Abbildung 9 – Umweltdiagramm.....	42
Abbildung 10 – Grobentwurf des Systems.....	42
Abbildung 11 – Übersichtsdiagramm Alfresco .....	44
Abbildung 12 – Übersichtsdiagramm L/information.....	45
Abbildung 13 – Grobentwurf Systemarchitektur.....	47
Abbildung 14 – Systemarchitektur .....	48
Abbildung 15 – Quelltextauszug: kategorieModel.xml .....	49
Abbildung 16 – Quelltextauszug: custom-model-context.xml.....	51
Abbildung 17 – Quelltextauszug: web-client-config-custom.xml.....	51
Abbildung 18 – Quelltextauszug: applicationModel.xml .....	52
Abbildung 19 – Quelltextauszug: newsModel.xml.....	54
Abbildung 20 – Struktur der Inhalte .....	55

Abbildung 21 – Verbindungsaufbau; Auszug aus Connector.java.....	57
Abbildung 22 – Abruf der Kategorien; Auszug aus Connector.java.....	58
Abbildung 23 – Abruf der Applikation; Auszug aus Connector.java .....	59
Abbildung 24 – UML-Diagramm Applikationen .....	60
Abbildung 25 – Aufbau der Fensterstruktur .....	62
Abbildung 26 – Konstruktor Navigation Panel; Auszug aus NavigationModel.java .....	63
Abbildung 27 – Darstellung der Applikationen; Auszug aus ApplicationOverviewModel.java.....	64
Abbildung 28 – Detailansicht einer Applikation .....	65
Abbildung 29 – L/information; Applikationsübersicht.....	65

# Abkürzungsverzeichnis

AIIM	Association for Information and Image Management
API	Application programming interface
BPM	Busines Process Management
CMIS	Content Management Interoperability Services
CMS	Content-Management-System
CRT	Content related technologie
ECM	Enterprise-Content-Management
ILOG	Informationslogistik
linX	logistic in eXellence
LTL	Lufthansa Technik Logistik GmbH
MRO	maintenance, repair, overhaul
MVC	Model, View, Control
OOA	Objektorientierte Analyse
PLOG	Physische Logistik
SLOG	Steuerungslogistik
UH/I	Abteilung Informationstechnologie
vpn	virtuell private network
XML	Extensible Markup Language

# 1. Einleitung

Ein effizienter und gezielter Informationsaustausch ist für die tägliche Arbeit in einem großen Unternehmen unerlässlich. Aus diesem Grund trägt die Kommunikation auf elektronischem Wege wesentlich zur gezielten und wirkungsvollen Informationsverteilung bei. In diesem Zusammenhang ist die Kommunikation und Verteilung von Informationen durch E-Mails zum festen Bestandteil des Büroalltags geworden. Einhergehend mit dieser Aussage belegen die Zahlen des Statistischen Bundesamts, dass 58% der deutschen Arbeitsplätze mit einem Computer ausgestattet sind und 46% dieser Arbeitsplätze über einen Internetzugang verfügen.<sup>1</sup> Der immense Anstieg der E-Mail-Kommunikation in Unternehmen bringt sowohl Vor- als auch Nachteile für die tägliche Arbeit. Die einfache Erstellung, der Versand an mehrere Empfänger und die direkte Zustellung auch in geografisch verteilten Unternehmen machen das Kommunikationsmedium E-Mail so leistungsstark. Durch die Vielzahl der täglich versendeten Mails in Unternehmen, ist es dem Nutzer meist nicht möglich alle Nachrichten sofort zu lesen oder zu bearbeiten. Daraus ergibt sich, dass die E-Mails nur selektiv vom Empfänger gelesen werden können, was zum größten Nachteil des Kommunikationsmediums führt: „unerwartete aber wichtige Nachrichten werden übersehen“<sup>2</sup>.

Zur Reduzierung der Informationsflut welche tagtäglich durch den E-Mail-Versand produziert wird, sowie zur Steigerung der internen Kommunikation, wird von der Lufthansa Technik Logistik GmbH (LTL) bereits seit Juni 2009 von der Abteilung Informationstechnologie ein berührungsempfindlicher Großbildmonitor zur Verbreitung von Informationen eingesetzt. Dieser Touchscreenmonitor ist zentral im Foyer der Abteilung angebracht und bietet allen Mitarbeitern die Möglichkeit die hinterlegten Informationen abzurufen. Hierbei handelt es sich um Informationen, welche für die Mitarbeiter wichtig sind, aber nicht den Anspruch verfolgen, dass sie persönlich in ein E-Mailpostfach zugestellt werden müssen. Ein weiterer Vorteil des Monitors ist neben der Reduzierung des E-Mail-Aufkommens, dass auch externe Dienstleister oder Kunden ohne betriebsinternes E-Mail-Postfach Zugang zu Informationen wie bspw. dem Mitarbeiterorganigramm erhalten. Derzeit wird auf dem Monitor eine PowerPoint Präsentation zur Informationsübermittlung dargestellt, wobei die Möglichkeit der Interaktion mit dem Nutzer durch die Touchfähigkeit des Monitors

---

<sup>1</sup> vgl. Soucek 2006, S. 59

<sup>2</sup> vgl. Soucek 2006, S. 59

heute ungenutzt bleibt. Weiterhin sind das Spektrum und die Menge der hinterlegten Informationen seit der Einführung 2009 derart angestiegen, dass die Informationsübermittlung durch die PowerPoint Präsentation erschöpft ist.

Die vorliegende Arbeit entwickelt ein Konzept zur effizienten und interaktiven Darstellung der Informationen auf dem Monitor. Besondere Berücksichtigung findet dabei die nachhaltige Pflege und Aufbereitung der Informationen. Die Realisierung des Konzepts als Prototyp bildet den Abschluss dieser Arbeit.

## 2. Einführende Grundlagen

### 2.1 Lufthansa Technik Logistik GmbH

Die vorliegende Arbeit wurde im Rahmen des Bachelorprojekts bei der Lufthansa Technik Logistik GmbH in Hamburg angefertigt. Die LTL wurde 1998 als 100-prozentige Tochtergesellschaft der Lufthansa Technik AG gegründet, welche wiederum dem Deutschen Lufthansa Konzern angehört. Die Lufthansa Technik Logistik GmbH ist weltweit vertreten und zählt mit 25 Niederlassungen in 21 Ländern zu den führenden Logistikdienstleistern in der Luftfahrtbranche.<sup>3</sup> Dabei reicht die Arbeit der LTL weit über die eines Spediteurs hinaus. Die schnellstmögliche und zuverlässige Transportabwicklung wie auch die sofortige Materialbeschaffung und die sachgerechte Lagerung und Verteilung von Ersatzteilen zählen zu den angebotenen Logistikdienstleistungen. Spezialtransporte, wie der Transport von Triebwerken oder Gefahrgut runden das Leistungsangebot der Lufthansa Technik Logistik GmbH ab.

### 2.2 Abteilung Informationstechnologie

Die Abteilung Informationstechnologie (UH/I) der LTL hält die Verantwortung zur durchgehenden Bereitstellung von IT-Applikationen, welche speziell auf die MRO-Logistik (MRO: **m**aintenance, **r**epair, **o**verhaul – de: Inspektion, Wartung, Instandsetzung) ausgerichtet sind. Diese, eigens für die Luftfahrtinstandhaltung in Bezug auf die Ersatzteil und Materialversorgung entwickelten Applikationen unterstützen die Logistikprozesse mit einzigartigen Funktionen mit dem Ziel, die volle Steuerung des gesamten Logistikprozesses zu erreichen. Zudem wird von der Abteilung UH/I ein aussagekräftiges Monitoring für die verschiedensten Operativbereiche bereitgestellt und die Fähigkeiten der Abteilung werden in zahlreichen Projekten entsprechend den Bedürfnissen der Kunden eingesetzt und weiterentwickelt. Die Informationstechnologieabteilung der LTL ist anhand der logischen und physischen Aufgabenbereiche in drei Gruppen unterteilt.<sup>4</sup>

---

<sup>3</sup> vgl. LTL, 2010

<sup>4</sup> vgl. Dr. Roth – LTL 2010

### **2.2.1 Informationslogistik (ILOG)**

Die Mitarbeiterinnen und Mitarbeiter der Gruppe Informationslogistik betreuen Applikationen speziell auf dem Gebiet der Informationsaufbereitung. Das bedeutet im Einzelnen, dass die Applikationen den physischen Logistikprozess und dessen Steuerung mit zusätzlichen Informationen oder Daten versorgen. Dies ermöglicht u.a. ein effektives und reichhaltiges Reporting oder Monitoring für kaufmännische Zwecke. In diesem Zusammenhang betreut die Gruppe ein Dokumentenmanagementsystem, verschiedene Tracking- und Tracing-relevante Applikationen sowie ein Archivierungssystem oder unterschiedliche Business-Intelligence-Anwendungen.

### **2.2.2 Physische Logistik (PLOG)**

Die Gruppe physische Logistik betreut und verantwortet alle Applikationen rund um die physischen Logistikprozesse der Lufthansa Technik Logistik GmbH. Diese Prozesse und die damit verbundenen Applikationen splitten sich in die verschiedensten Tätigkeitsfelder wie Wareneingang, Warehouse Management und Versand, Zollverfahrensermittlung, Warenbestandsführung sowie die Dokumentation und Abrechnung aller externen Transporte auf. Sowohl der sichere, stabile Betrieb als auch die nachhaltige fachliche Entwicklung der Applikationen gehören zu den Aufgaben der Gruppe PLOG wie auch die Durchführung von Projekten beginnend bei Konzeption und Realisierung über Tests und Abnahme bis hin zur Produktivsetzung und Optimierung. Zusätzlich betreut die Gruppe physische Logistik im Team Customer Solutions die applikationsübergreifende Aufnahme, Abstimmung und Umsetzung aller IT-bezogenen Anforderungen für externe Kundenprojekte.

### **2.2.3 Steuerungslogistik (SLOG)**

Im Rahmen der Planung und Durchführung von IT-Projekten, die sich mit der Steuerung und zeitlichen / prozessualen Planung von innerbetrieblichen Logistikprozessen beschäftigen, steht die Gruppe Steuerungslogistik den internen und externen Kunden sowohl national als auch im internationalen Bereich zur Seite. Zusätzlich verantwortet die Gruppe SLOG die IT-seitige Planung und Steuerung der internen und externen Transporte. In diesem Zusammenhang betreut die Gruppe das zentrale Auftragsmanagementsystem linX (logistic in eXcellence), welches als zentraler Eingangskanal für jegliche Logistikaufträge fungiert und über Schnittstellen in permanenter Kommunikation mit den operativen



Systemen steht. Weitere Ausführungen zur Applikation linX sind dem Praxisbeleg zu entnehmen.

## **2.3 Interne Kommunikation**

Die Komplexität und Vielschichtigkeit der Abteilung Informationstechnologie bedeutet gleichermaßen hohe Verantwortung in Bezug auf die interne Kommunikation. Wie bereits eingangs erwähnt ist durch die weite Verbreitung von PC-Arbeitsplätzen verbunden mit E-Mail, Intranet und Internet die Kommunikation erheblich vereinfacht und gleichzeitig anspruchsvoller geworden. Damit einhergehend sind in vielen Unternehmen die Abteilungen für Öffentlichkeitsarbeit viel zu sehr darauf bedacht, das Ansehen der Firma nach außen zu pflegen und zu organisieren. Die interne Kommunikation im Unternehmen wird für die folgenden Betrachtungen als eine Art der Kommunikation verstanden, mit deren Hilfe Ereignisse und Nachrichten durch geeignete Kommunikationswege vom Management bis in alle Bereiche des Unternehmens verbreitet werden.<sup>5</sup> In der allgemeinen Unternehmerwelt gilt die interne Kommunikation jedoch als vernachlässigbare Größe, dabei ist gerade die Information und Aufklärung der Mitarbeiter ein wichtiges Instrument für ein erfolgreiches Unternehmen und führt zu der Schlussfolgerung, dass die Kommunikation nicht nur außerhalb sondern auch innerhalb des Unternehmens wirken muss, wenn sie funktionieren soll.<sup>6 7</sup> Warum es genauso wichtig ist, die Mitarbeiter mindestens ebenso gut zu informieren wie die Öffentlichkeit und mithilfe welcher Kommunikationswege dies geschehen kann, wird im Folgenden allgemein und anschließend in Bezug auf die Abteilung Informationstechnologie der Lufthansa Technik Logistik näher betrachtet.

### **2.3.1 Notwendigkeit der internen Kommunikation**

Die Arbeitsmarktsituation hat sich im Vergleich zu den vergangenen Jahren sukzessive verändert. Immer seltener sind Werk tätige vom Berufseinsteiger bis hin zur Rente in einem Unternehmen tätig. Umfangreiches und gezieltes Wissen über ein Unternehmen, welches sich ein Mitarbeiter über mehrere Berufsjahre oder gar –jahrzehnte erarbeitet hat und damit eine gewisse Verbundenheit mit dem Unternehmen, geht durch häufiges wech-

---

<sup>5</sup> vgl. Künzel 2005, S. 193

<sup>6</sup> vgl. Künzel 2005, S. 193ff

<sup>7</sup> vgl. Puttenat 2007, S. 111f

seln des Arbeitsplatzes verloren.<sup>8</sup> Doch nicht allein der Arbeitsmarkt und die damit verbundene Fluktuation der Mitarbeiter machen eine interne Informationsgestaltung erforderlich, sondern auch die heutige Wirtschaftslage. Unternehmensfusionen, Wechsel der kompletten Führungsebene, Stellenabbau oder der Börsengang eines Unternehmens sind nur einige Beispiele weswegen die interne Kommunikation wichtiger ist denn je.<sup>9</sup>

## **2.3.2 Arten der unternehmensinternen Kommunikation**

Die interne Kommunikation in großen und mittelständischen Unternehmen ist einem Wandel unterlaufen. Die Zeit, in der der Chef seine Mitarbeiter über Rundschreiben informierte, ein schwarzes Brett für Neuigkeiten oder Änderungen im Unternehmen existiert, gehören meist der Vergangenheit an. Der Grund für diesen Übergang ist gleichermaßen die veränderte Kommunikation durch die Entwicklung der Computer, das Internet sowie der verstärkte Einsatz von Firmenhandys verbunden mit dem E-Mailverkehr.<sup>10</sup> Zwei ausgesuchte Beispiele für die interne Kommunikation mittels neuer und herkömmlicher Medien werden nachfolgend kurz betrachtet sowie deren Vor- und Nachteile beschrieben.

### **2.3.2.1 Intranet**

Das Intranet ist in vielen Unternehmen zu einer der wichtigsten Anlaufstellen bezüglich unternehmensinterner Informationen geworden. Es kann als firmeneigenes „Online-Portal“<sup>11</sup> verstanden werden. Das bedeutet im Einzelnen, dass das Intranet eines Unternehmens von firmeninternen Arbeitsplätzen aufgerufen werden kann. Es ist also losgelöst vom Internet im Unternehmen verfügbar. Weiterhin besteht die Möglichkeit, mithilfe einer vpn-Verbindung (**v**irtuell **p**rivate **n**etwork), welche von einem Arbeitsplatz außerhalb der Firma aufgebaut wird, auf das Unternehmensnetzwerk und somit auch auf das Intranet zuzugreifen. Im Intranet können die verschiedensten Inhalte dargestellt werden: Neuigkeiten, Arbeitszeitrachweise, Organigramme oder Verweise auf unternehmensinterne Webapplikationen sind nur einige Beispiele. Das Intranet gehört zu einem schnellen Medium, was bedeutet, dass die hinterlegten Informationen fortwährend aktualisiert und aufbereitet werden müssen, um die Akzeptanz unter den Mitarbeitern sicherzustellen. In diesem Zusammenhang ist die Abteilung für interne Kommunikation sowie die EDV-

---

<sup>8</sup> vgl. Puttenat 2007, S. 112

<sup>9</sup> vgl. Puttenat 2007, S. 112

<sup>10</sup> vgl. Puttenat 2007, S. 111

<sup>11</sup> vgl. Puttenat 2007, S. 118

Abteilung eines Unternehmens eng miteinander verbunden. Erstere liefert die entsprechenden Inhalte welche von der EDV-Abteilung technisch aufbereitet und im Intranet zur Verfügung gestellt werden.<sup>12</sup> Die Deutsche Lufthansa AG bietet allen seinen Mitarbeitern ein einheitliches Firmenintranet. Das so genannte „eBase“ wird zentral verwaltet und bietet Informationen, Applikationen, Formulare oder Hintergrundwissen für alle Konzernbereiche und Gesellschaften. Die Vorteile eines Firmenintranets sind die ständige Aktualität sowie die Möglichkeit, ein enorm reichhaltiges und umfangreiches Informationsangebot für ein ganzes Unternehmen zu schaffen. Nachteilig hingegen ist der hohe Pflegeaufwand, der sich ergibt, um fortwährend aktuelle Informationen anzubieten. Weiterhin ist das Intranet meist nur innerhalb des Unternehmens verfügbar. Operativen Mitarbeitern ohne eigenen Rechnerarbeitsplatz fehlt gegebenenfalls der Zugriff auf das Firmenintranet.

#### **2.3.2.2 Mitarbeiterzeitschrift**

Ein weiteres Medium zur Verteilung von firmeninternen Informationen und Mittel der internen Kommunikation ist die Mitarbeiterzeitschrift. In der heutigen Zeit, in der Unternehmensmitteilungen im Intranet nur einen Mausklick entfernt sind, scheint die herkömmliche Mitarbeiterzeitung als überholt. Dem ist jedoch nicht so, was folgende Eigenschaften einer Mitarbeiterzeitschrift zeigen. Sie ist zum einen im ganzen Unternehmen verfügbar – kann am persönlichen Arbeitsplatz oder in der Kantine gelesen werden – und ist somit auch für Mitarbeiter ohne Zugang zum Firmenintranet greifbar. Zum anderen handelt es sich, gegeben durch das Printmedium, um beständige Informationen, welche erhalten bleiben. Die Zeitschrift kann losgelöst von Unternehmensgrenzen eingesetzt werden und involviert Familie und Freunde in das Arbeitsleben. Aus diesem Grund bietet die Mitarbeiterzeitung eine gute Kommunikationsmöglichkeit und steigert das „wir“-Gefühl der Mitarbeiter. Die Mitarbeiterzeitung der Lufthansa AG erscheint jede Woche neu und erreicht weltweit mit einer Auflage von 75.000 Zeitschriften mehr als 90.000 Lufthansa Mitarbeiter in zwei Sprachen. Nachteilig bleibt anzumerken, dass auch für eine Mitarbeiterzeitschrift eine eigene Redaktion notwendig ist. Dies macht eine evtl. Einführung, Unterhaltung oder Publikation der Zeitschrift für kleine oder mittelständische Unternehmen aufwändig.<sup>13</sup>

---

<sup>12</sup> vgl. Puttenat 2007, S. 118ff

<sup>13</sup> vgl. Puttenat 2007, S. 122ff

## **2.4 Aufgabenstellung**

### **2.4.1 Motivation**

Seit Juni 2009 wurden in der Abteilung Informationstechnologie (UH/I) der Lufthansa Technik Logistik GmbH zahlreiche Maßnahmen zur Optimierung der internen und externen Präsenz erreicht. Ziel war es, sowohl internen als auch externen Mitarbeitern, sowie Kunden und Dienstleistern die Arbeitsabläufe näher zu bringen und ein besseres Verständnis der Unternehmensprozesse zu erreichen. Um den stetigen Kommunikationsfluss zu gewährleisten, sollen umfassende, zeitgemäße Informationen zur Verfügung gestellt werden, die einerseits leicht zu handhaben und andererseits leicht zugänglich sind. Im Zuge dessen wurde in der Abteilung ein berührungsempfindlicher Großbildmonitor an einem zentralen Standort angebracht, um Informationen aktuell zu vermitteln, anschaulich zu gestalten und interaktiv zu präsentieren.

Gegenwärtig wird auf dem Monitor eine PowerPoint Präsentation dargestellt, welche zyklisch Informationen zu verschiedenen Themen wie Applikationsverfügbarkeiten in der Abteilung, geplante Auszeiten, personelle Veränderungen bei UH/I, Success Stories oder Nachrichten über die Lufthansa Technik Logistik GmbH, die Lufthansa Technik AG oder die Deutsche Lufthansa zeigt. Für einen zeitgemäßen Informationsstand wird diese Präsentation wöchentlich aktualisiert. Die berührungsempfindliche Oberfläche des Monitors wird bislang noch nicht verwendet. In diesem Zusammenhang wurde von der Arbeitsgruppe für Kommunikation und Organisation ein sogenanntes Touchscreen Konzept entwickelt, welches die derzeitige Monitor-Präsentation mit all Ihren Inhalten strukturiert und die Darstellung für ein neues System beschreibt. Dieses Touchscreen Konzept ist u.a die Grundlage für die technische Umsetzung.

### **2.4.2 Zielstellung**

Zur effizienten Nutzung des Monitors ist ein IT-Konzept zu entwickeln, welches die Darstellung der Inhalte gemäß des Touchscreen Konzepts und die Interaktion mit dem Anwender ermöglicht. Der besondere Fokus liegt dabei auf einer effektiven und einfachen Datenspeicherung und -pflege der Informationen. Eine prototypische Implementierung ist nach Abschluss der theoretischen Betrachtungen durchzuführen, um die systemseitige Umsetzung zu demonstrieren. In diesem Zusammenhang wird der Entwicklungsprozess mit den Schwerpunkten Systemmodellierung und -beschreibung betrachtet. Anschließend

wird eine Dokumentation mit Hilfe verschiedener UML-Standards auf Basis des Prototyps vorgestellt. Aufgrund der komplexen und vielseitigen Einsatz- und Anwendungsmöglichkeiten des Softwareprodukts in Bezug auf die internen Kommunikationsinhalte werden Weiterentwicklungsmöglichkeiten sowie zukünftige Einsatzgebiete betrachtet und bilden den Abschluss dieser Arbeit.

Losgelöst von den Anforderungen an die Software werden aufgrund der momentan zur Darstellung verwendeten Präsentation folgende zwei Hauptanforderungen an das zu entwickelnde IT-Konzept gestellt. Einerseits, ist eine Möglichkeit zur nachhaltigen Pflege und Aufbereitung der Informationen losgelöst von verschiedenen Darstellungsformen zu finden. In zweiter Hinsicht ist wegen des großen Informationspools eine einfache Navigation zu konzipieren, die dem Anwender eine einfache Datenabfrage ermöglicht.

### **2.4.3 Abgrenzung**

Die vorliegende Arbeit beschreibt die konzeptionelle Entwicklung des Prototyps zur internen Kommunikation mittels des berührungsempfindlichen Monitors der Abteilung Informationstechnologie der Lufthansa Technik Logistik GmbH. Wie in der Softwareentwicklung üblich, ist ein Lasten- und Pflichtenheft nicht Gegenstand dieser Arbeit, da die Anforderungen an das Softwareprodukt in verschiedenen Besprechungen verbal definiert wurden. Aufgrund der Realisierung als Prototyp zur Demonstration der Umsetzungsmöglichkeit fließt ferner keine Aufwandsschätzung zur Implementierung des Softwareproduktes in die vorliegende Arbeit ein.

### 3. Softwareauswahl

Die Entwicklung und Einführung eines Softwareprodukts zur internen Kommunikation mithilfe des Touchscreen Monitors in der Abteilung Informationstechnologie bedarf der Betrachtung verschiedener Umsetzungsvarianten. Im Zuge dessen werden im Folgenden verschiedene informationstechnische Lösungen beschrieben und evaluiert. Anschließend wird, unter Berücksichtigung der gewonnen Erkenntnisse sowie der Vorgaben der Arbeitsgruppe Organisation und Kommunikation der LTL, eine Entscheidung getroffen.

#### 3.1 Wege der Softwareeinführung und -entwicklung

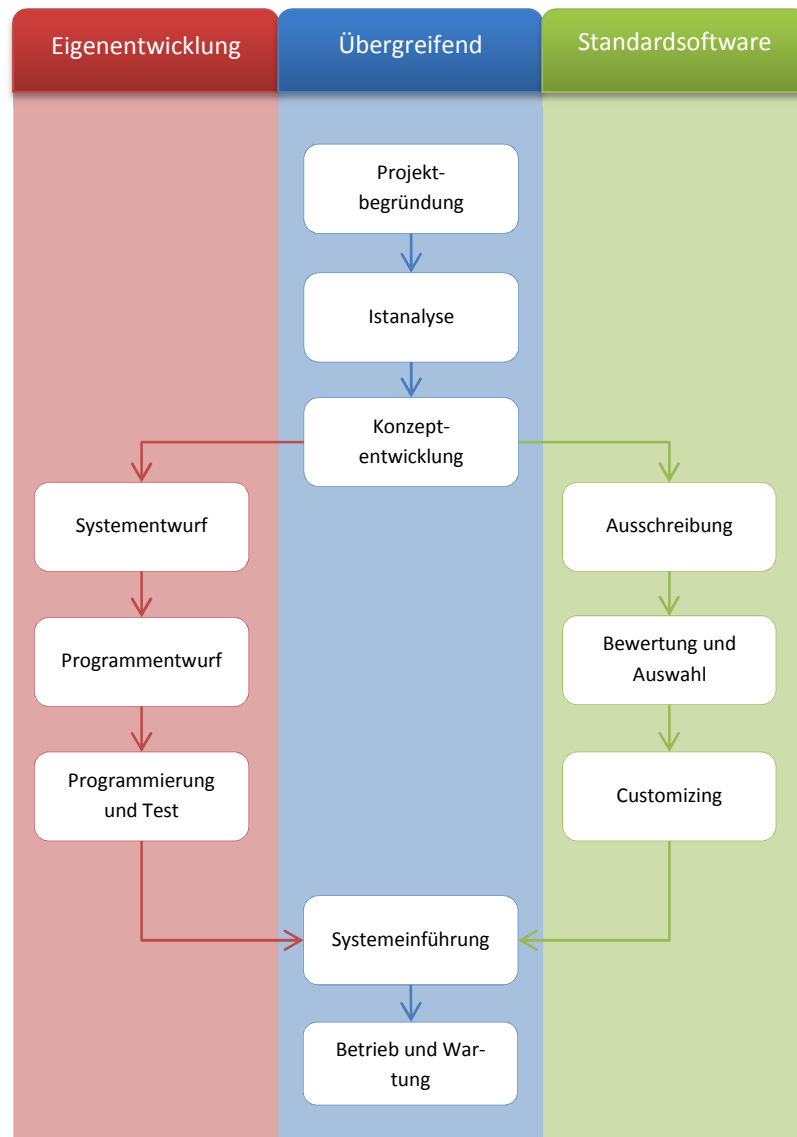
Zur Planung und Realisierung von Informationssystemen existieren zwei grundsätzliche Varianten in der Umsetzung. Es besteht die Möglichkeit, das Softwareprodukt in Eigenentwicklung oder aber durch die Auswahl und Einführung von Standardsoftware zu realisieren.<sup>14</sup> Dabei ist es möglich, beide Formen weiter zu differenzieren. Dies bedeutet im Einzelnen, dass eigenentwickelte Software entweder von eigenen Mitarbeitern entwickelt (interne Lösung), oder aber durch ein externes Dienstleistungsunternehmen realisiert wird (externe Lösung). Beim Eingangs erwähnten zentralen Auftragsmanagementsystem linX handelt es sich bspw. um eine Individualsoftware, die durch den externen Dienstleister SALT Solutions umgesetzt wurde. Bei der Auswahl und Einführung von Standardsoftware existieren ebenso zwei grundlegende Handlungsalternativen. Einerseits besteht die Möglichkeit die Software zu kaufen und durch die eigenen Mitarbeiter einzuführen. Auf der anderen Seite können Standardsoftwareprodukte gemietet werden. Die Software wird demzufolge durch den Softwareanbieter eingeführt und administriert. Im direkten Vergleich zur Individualsoftware stellt der Kauf von Standardsoftware die interne und das Mieten die externe Lösung dar.<sup>15</sup> Folgende Grafik beschreibt das allgemeine Vorgehensmodell zur Einführung eines Softwareprodukts für die internen Varianten. In diesem Zusammenhang sind Unterschiede der vorgestellten Handlungsalternativen, die sich durch die Entwicklung von Individualsoftware oder der Auswahl und Einführung von Standardsoftware ergeben, zu erkennen.

---

<sup>14</sup> vgl. Gadatsch 2010, S. 321f

<sup>15</sup> vgl. Gadatsch 2010, S. 322

Abbildung 1 – Vorgehensmodell



Quelle: in Anlehnung an: Hasenkamp, Stahlknecht 2010, S. 221

### 3.1.1 Einführung von Standardsoftware

Ein Standardsoftwaresystem ist ein betriebliches Anwendungssystem, welches aufgrund seiner Funktionsabdeckung im Hinblick auf betriebswirtschaftliche oder technische Gegebenheiten von vielen Unternehmen einer oder mehrerer Branchen verwendet werden kann.<sup>16</sup> Wie in Abbildung 1 beschrieben, ist eine Ausschreibung zu erstellen. Diese beinhaltet konkrete Forderungen an ein Softwareprodukt. Die Rückmeldungen verschiedener Anbieter von Standardsoftware sind anschließend in Bezug auf Funktionsabdeckung und

<sup>16</sup> vgl. Weder 2003, S. 22

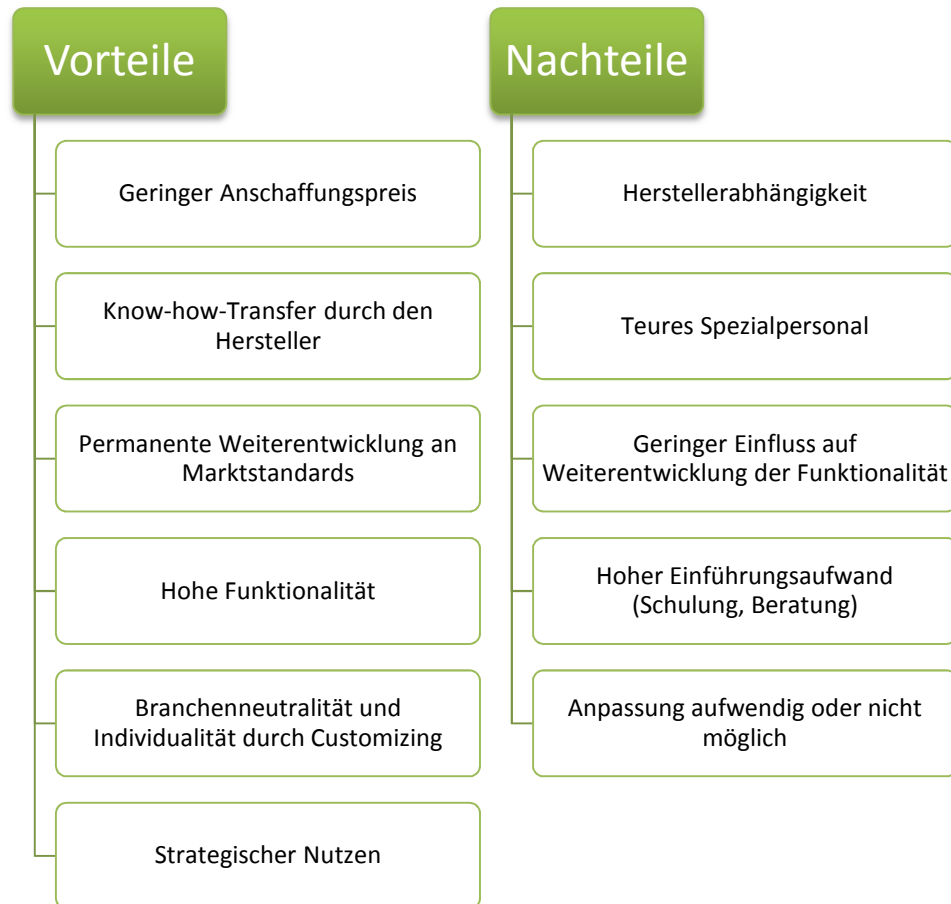
der unternehmerischen Prozessabbildung zu bewerten. Selbstverständlich spielen auch ökonomische Gesichtspunkte eine Rolle. Vor der Systemeinführung ist die Software ggf. entsprechend anzupassen und zu konfigurieren. Dabei können die Entwicklungshäuser von Standardsoftware meist auf jahrelange Erfahrung und technisches Know-How zurückgreifen. Durch die Anwendung eines Standardsoftwareprodukts erhalten die Unternehmen die Möglichkeit, auf „Best-practice“ (= korrekter betriebswirtschaftlicher oder technischer Ablauf) zurückzugreifen. Weiterhin ist es möglich, die Standardsoftware durch Parameter an individuell im Unternehmen ablaufende Prozesse anzupassen. Man spricht vom Customizing.<sup>17</sup> Über das Customizing hinausgehend, bieten Standardsoftwareprodukte wenig Raum für spezielle Anpassungen wie bspw. individuelle optische Darstellung nach firmeninternen Corporate Design Vorschriften. Weitere Nachteile sind die Abhängigkeit zum Hersteller der Software bei Änderungswünschen sowie ein hoher Einführungsaufwand. Folgende Abbildung zeigt die Vor- und Nachteile von Standardsoftware im Überblick.

---

<sup>17</sup> vgl. Weder 2003, S. 22f



Abbildung 2 – Vor- und Nachteile von Standardsoftware

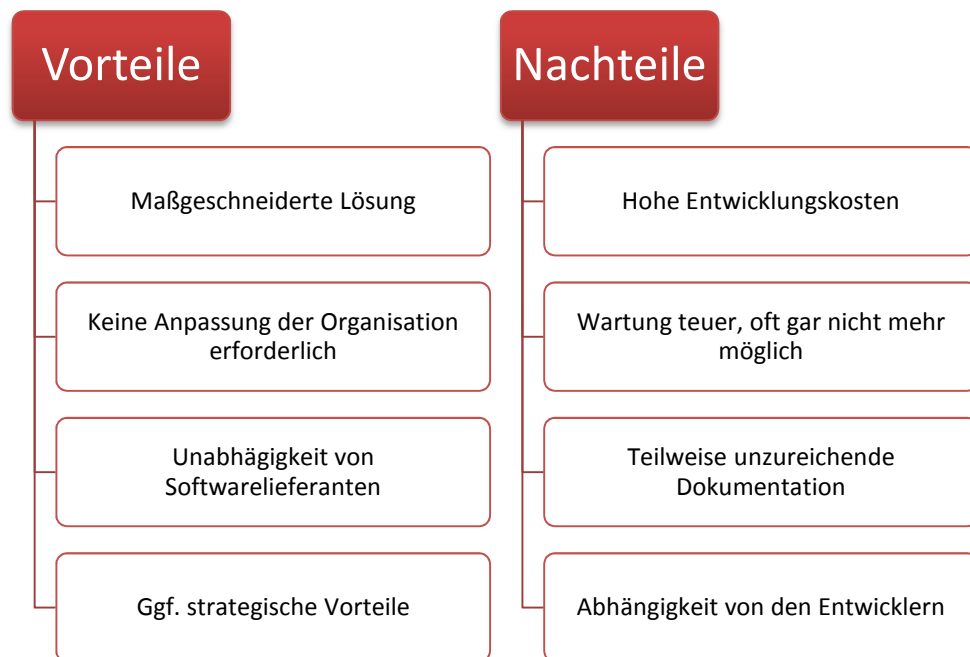


Quelle: Gadatsch 2010, S. 327

### 3.1.2 Eigenentwicklung des Softwareprodukts

Analog zur Ausschreibung bei Standardsoftware, sind für ein individuelles Softwareprodukt ein Systementwurf und anschließend ein Programmmentwurf zu erstellen. Anhand dieser wird die Software im Anschluss programmiert und fortwährend auf die Funktionsfähigkeit getestet. Wie bereits bei der Einführung von Standardsoftware betrachtet, existieren auch bei der eigenständigen Entwicklung von Software diverse Gründe, die für oder gegen diese sprechen. Nachstehende Grafik zeigt eine Zusammenfassung dieser Vor- und Nachteile.

Abbildung 3 – Vor- und Nachteile von Individualsoftware



Quelle: Gadatsch 2010, S. 325

Die Erstellung und Einführung von Individualsoftware in Eigenentwicklung ermöglicht es dem Unternehmen, das Softwareprodukt mit allen erforderlichen Anforderungen zu erstellen. Die Software muss dementsprechend nicht an unternehmensspezifische Prozesse adaptiert werden, da sie speziell auf das Unternehmen zugeschnitten ist. Des Weiteren muss die unternehmensinterne Organisation nicht zwingend verändert werden, da das Softwareprodukt genau so gestaltet werden kann, dass es sich möglichst optimal in bestehende Organisationsstrukturen einbinden lässt. Die strategischen Vorteile, die durch den Einsatz von Individualsoftware entstehen können, schaffen dem Unternehmen einen Wettbewerbsvorteil und gelten somit als Hauptargument ein Softwareprodukt eigens zu entwickeln.<sup>18</sup> Dies verdeutlicht ein Beispiel der Lufthansa Technik Logistik GmbH. Das in Eigenentwicklung implementierte Auftragsmanagementsystem linX verknüpft aufwendige Terminierungsverfahren mit Kostenfaktoren, um die Ersatzteilversorgung zur richtigen Zeit am richtigen Ort zu gewährleisten. Dieses Terminierungsmodell verschafft dem Unternehmen besonders bei zeitkritischen Materialanforderungen einen erheblichen Vorteil.

Dem gegenüberstehend sind jedoch die Nachteile von Individualsoftware anzumerken. Stehen dem Softwareprojekt nicht genügend qualifizierte Entwickler zur Verfügung, so

<sup>18</sup> vgl. Gadatsch 2010, S. 325f

sind die Entwicklungskosten höher als die beim Kauf von Standardsoftware anfallenden Kosten. Weiterhin ist trotz gut organisierten Praxisprojekten das Zeitmanagement in den meisten Fällen kritisch. Dies ist ein möglicher Grund, weswegen die Dokumentation des Softwarevorhabens vernachlässigt wird. Schließlich ist Software, welche ausführlich dokumentiert ist, nicht leistungsfähiger. Dadurch ist eine gewisse Abhängigkeit zu den Entwicklern des Softwareprodukts gegeben.<sup>19</sup>

## 3.2 Auswahl der Umsetzungsvariante

Zur Auswahl der passenden Umsetzungsvariante sind neben strategischen Unternehmensgrundsätzen zwei weitere Faktoren von Bedeutung, um entweder Standardsoftware oder Individualsoftware zu bevorzugen. Zu beachten ist, dass in den meisten Softwareprojekten, abhängig vom Produkteinsatz, Standardsoftware zum Teil mit Individualsoftware ergänzt wird (über das Customizing hinausgehend) und auch Individualsoftwareprodukte durch die Anwendung von Standardsoftware unterstützt werden können. Dabei richtet sich die Entscheidung nach der Verfügbarkeit von Standardsoftwareprodukten und die von der Software ausgehende strategische Bedeutung. Dadurch ergibt sich ein Entscheidungsportfolio, welches in vier verschiedene Felder unterteilt ist.<sup>20</sup> Das in der Literatur beschriebene Portfolio wird im Folgenden näher betrachtet und durch ausgewählte Applikationen der Lufthansa Technik Logistik angereichert. Abschließend wird das Entscheidungsportfolio auf das Softwarevorhaben zur internen Kommunikation angewendet.

---

<sup>19</sup> vgl. Gadatsch 2010, S. 326

<sup>20</sup> vgl. Gadatsch 2010, S. 326

Abbildung 4 – Eigenentwicklung vs. Standardsoftware



Quelle: Gadatsch 2010, S. 323

Wie in Abbildung 4 erkennbar ist ein Softwareprodukt, welches in standardisierter Form nicht oder nur geringfügig am Markt zu finden, jedoch aber von hoher strategischer Bedeutung ist, in Eigenentwicklung zu realisieren. Die Auftragsmanagementsoftware linX forderte spezielle Anforderungen, gegeben durch den Einsatz in der Luftfahrtindustrie. Sie ist in der Lage, von verschiedensten Airlines dieser Welt Aufträge entgegenzunehmen und die Applikation beherrscht die Zuordnung der Aufträge zu einer Prozessvariante. Zudem wendet die Software verschiedenste Terminierungsalgorithmen an, um jeden operativen Bereich zur richtigen Zeit zu beauftragen. Zusammengefasst stellt linX eine Anwendung dar, die als Standardsoftware schlecht am Markt verfügbar ist und gleichzeitig eine hohe strategische Bedeutung für das Unternehmen darstellt. linX wurde 2004 in Eigenentwicklung durch einen externen Dienstleister entwickelt. Dies ermöglichte es, das Softwareprodukt genau nach den Anforderungen der Lufthansa Technik Logistik GmbH zu entwickeln.<sup>21</sup> Im Entscheidungsportfolio ist diese Art der Umsetzung als *Make* gekennzeichnet

Ferner zeigt die Abbildung, dass Softwareprojekte, die einen geringen strategischen Stellenwert einnehmen und zudem als Standardsoftwareprodukt verfügbar sind, durch den Kauf von Software umzusetzen sind. Die physischen Prozesse des Ersatzteillagers der

<sup>21</sup> Musch 2010, S.5ff

LTL entsprechen mit wenigen Ausnahmen denen einer Standardlagerverwaltung. Lagerverwaltungssysteme sind gut als Standardsoftware am Markt erhältlich und die strategische Bedeutung eines solchen Systems ist eher gering. Das Lagerverwaltungssystem der LTL wurde als Standardsoftware gekauft und durch die Customizing-Möglichkeit entsprechend angepasst. *Buy and Customize*.

Das Softwareprodukt zur internen Kommunikation der Abteilung Informationstechnologie, welches durch diese Arbeit entworfen wird, ist von mittlerer strategischer Bedeutung. Standardisierte Software ist aufgrund der speziellen Anforderungen des Touchscreen-Konzepts nicht verfügbar, weswegen die Software als Individualsoftware entwickelt wird. Zu einem großen Teil wird das eigenentwickelte Softwareprodukt durch standardisierte Software unterstützt. Hierbei könnte es sich bspw. um Standardsysteme zur Datenpflege und –speicherung handeln, welche die eigentliche Applikation unterstützen. In diesem Zusammenhang wird im Folgenden das Content Management sowie damit verbundene IT-Systeme betrachtet und auf deren Eignung als Datenpool zur Ablage der Inhalte zur internen Kommunikation untersucht.

## 4. Content Management

Das Content Management umfasst alle Aktivitäten zur Planung, Verwaltung, Steuerung und Koordination von verschiedenen Inhalten (engl. Content).<sup>22</sup> In Bezug auf die Informationen zur internen Kommunikation handelt es sich bspw. um Neuigkeiten, Organigramme oder Statistiken. In diesem Kapitel werden grundlegende Content Management Technologien (Content Related Technologie – CRM) vorgestellt. Anschließend werden diese bezogen auf IT-Systeme angewendet und sollen die Grundlage bilden, den Informationscontent zur internen Kommunikation in einem Datenpool bereitzustellen.

Für die konkrete Definition und Modellierung eines Systems, welches den Informationscontent bereitstellt, ist es erforderlich, grundlegende Zusammenhänge über Informationen zu erläutern. In der digitalen Welt wird der Inhalt aus verschiedenen Informationen zusammengesetzt – man spricht von der „Anatomie der Informationen“<sup>23</sup>. Diese Hauptkomponenten der Informationen sind die *Struktur*, die *Darstellung* und der eigentliche *Inhalt* und sind u.a. Voraussetzung für die automatisierte Weiterverarbeitung und Wiederverwendung.<sup>24</sup> Der Zusammenhang dieser einzelnen Komponenten sind in den Content Related Technologien verankert und finden in verschiedenen IT-Systemen Anwendung. Als Beispiele für diese Systeme sind Content-Management-Systeme (CMS), Enterprise-Content-Management-Systeme (ECMS) aber auch Business-Process-Management-Systeme (BPM) zu nennen.<sup>25</sup> In diesem Zusammenhang wird im Folgenden speziell auf CMS und ECMS eingegangen.

### 4.1 Content-Management-System

Ein Content-Management-System wird vereinfacht dargestellt für die Verwaltung verschiedenster Medienarten (bspw. Texte, Bilder, Videos) sowie deren Präsentation eingesetzt.<sup>26</sup> Detailliert betrachtet, entwickelten sich aus Content-Management-Systemen verschiedene Ableger, welche in Verbindung mit speziellen Arten von Inhalten Anwendung

---

<sup>22</sup> vgl. Riggert 2009, S. 1f

<sup>23</sup> vgl. Schuster, Wilhelm 23.12.2000, S. 373

<sup>24</sup> vgl. Schuster, Wilhelm 23.12.2000, S. 373

<sup>25</sup> vgl. Schmitz 2007, S. 210

<sup>26</sup> vgl. Schmitz 2007, S. 210

finden. Zu nennen sind bspw. Dokumente in Dokumenten-Management-Systemen oder Webinhalte in Web-Content-Management-Systemen.<sup>27</sup> Für die folgenden Betrachtungen wird die Art des Content zunächst vernachlässigt und das Content-Management-System als Oberbegriff eingesetzt. Die mit einem Content-Management-System verbundene Anwendung der CRT führt zu einem erheblichen Mehrwert. Um dies zu belegen wird zunächst der Zusammenhang von Strukturierung und Darstellung der Inhalte beschrieben. Anschließend wird speziell auf den daraus resultierenden Nutzen eingegangen. Zum Schluss wird der Einsatz eines solchen Systems zur Erfüllung der mit dieser Arbeit verbundenen Ziele bewertet.

#### **4.1.1 Strukturierung und Darstellung der Inhalte**

Die Strukturierung des Contents gilt als die wichtigste Komponente der eigentlichen Information. In diesem Zusammenhang sind zum einen die Metadaten, also Zusatzinformationen über das Objekt, zu erwähnen, mit deren Hilfe Content strukturiert werden kann. Kurz gesagt sind Metadaten Daten über Daten. Beispiele hierfür sind das Aufnahmedatum eines Bildes oder die Autoren einer Textdatei. Bezogen auf die CRT werden diese Metadaten verwendet, um den Content zu identifizieren. Als Beispiel dienen zwei Texte, deren Schriftgröße, Schriftart und Schriftstil vollkommen identisch ist. In den Metadaten des einen ist jedoch vermerkt, dass es sich bei einem um eine Überschrift und bei dem anderen um eine Kurzbeschreibung handelt. Anhand dieser Zusatzinformationen sind Systeme in der Lage, zwei gleichartige Texte zu differenzieren und zu verarbeiten.<sup>28</sup> Die zweite Art der Strukturierung des Contents ist die Gruppierung in hierarchischer Anordnung. Dabei kann man sich die Hierarchie als eine Art Baumstruktur vorstellen. Auf diese Weise ist es bspw. möglich, alle Berichte einer Fachzeitschrift systematisch unter dieser anzuordnen und folglich sind die einzelnen Artikel eindeutig einem übergeordneten Content zugewiesen.<sup>29</sup> Werden beide Arten der Strukturierung kombiniert, ergeben sich demzufolge verschiedenste Möglichkeiten die Art und Zugehörigkeit des Contents klar und logisch zu definieren.

Die zweite Komponente der Information, nämlich deren Darstellung oder auch Formatierung ist in enger Verbindung mit der Struktur zu sehen. Betrachten wir bspw. wieder den Artikel einer Fachzeitschrift so ist dessen Struktur für den Betrachter eindeutig durch ver-

---

<sup>27</sup> vgl. Schmitz 2007, S. 209ff

<sup>28</sup> vgl. Schmitz 2007, S. 199f

<sup>29</sup> vgl. Schmitz 2007, S. 200f

schiedene Formate, wie Schriftgrößen oder Absätze definiert und auf Anhieb ersichtlich. In einigen Fällen kann es dabei vorkommen, dass der Inhalt ausschließlich über die Formatierung oder die Darstellung strukturiert ist. In diesem Zusammenhang kann der Artikel der Zeitschrift beliebig oft vervielfältigt und auf andere Printmedien übertragen werden. Die automatische Überführung des Contents auf andere Medien, wie bspw. einer Webseite, ist jedoch nicht möglich, da die Struktur nicht in dem Content verankert ist.<sup>30</sup> Folglich muss für eine automatisierte Verwertung von Inhalten eine explizite Strukturierung vorgegeben werden, auf deren Basis die Formatierung erstellt wird. Dieser beschriebene Sachverhalt wird dementsprechend zur Strukturierung der Inhalte zur internen Kommunikation besonders berücksichtigt. Die Anwendung der vorgestellten Strukturierung, welche zur automatischen Verarbeitung notwendig ist, birgt zudem noch weitere Vorteile, die im folgenden Kapitel vorgestellt werden.

#### 4.1.2 Nutzenpotentiale

Ein Content-Management-System ist ein System in dem die Content Related Technologien Anwendung finden. Im vorherigen Kapitel wurde bereits kurz auf die automatische Verwendung von Inhalten in einem Printmedium und auf einer Webseite eingegangen. Man spricht in diesem besonderen Fall von der *Mehrfachverwertung* des Contents.<sup>31</sup> Diese Mehrfachverwertung ist nicht nur aus informationstechnischer Sicht ein erheblicher Vorteil, sondern auch aus ökonomischer. Jedem Content kann ein definierter Wert zugewiesen werden, welcher meist in enger Beziehung mit den bestehenden Rechtsverhältnissen betrachtet wird. Vordergründig handelt es sich dabei um geltendes Urheberrecht für einen bestimmten Content, wie bspw. einen wissenschaftlichen oder literarischen Text.<sup>32</sup> Die Möglichkeit der Wiederverwertung eines solchen Contents führt folglich zu Kosteneinsparungen für das Unternehmen. Aber nicht nur die Wiederverwertung von urheberrechtlich geschützten Werken führt zu Kosteneinsparung, sondern auch, weil bereits angelegte Inhalte nicht neu produziert werden müssen, um sie für andere Ausgabekanäle verfügbar zu machen.

Weiterhin stehen dem Anwender durch den Einsatz von Content Related Technologie Systemen viele Funktionen zur effizienten Erfassung und Verwaltung der Inhalte zur Verfügung. Die Funktionen reichen dabei bspw. von automatischen Versionierungsmethoden,

---

<sup>30</sup> vgl. Schmitz 2007, S. 201

<sup>31</sup> vgl. Schmitz 2007, S. 204

<sup>32</sup> vgl. Schmitz 2007, S. 199f



über Archivierungsverfahren bis hin zu komplexen Suchalgorithmen. Diese verschiedenen Funktionalitäten bieten dem Anwender einen so genannten *Funktionalitätsgewinn*.<sup>33</sup>

Zusammenfassend steigern CRT Systeme die *Effizienz* im Umgang mit verschiedenen Arten von Inhalten durch die strukturierte Erfassung, die Integration verschiedener Quellen oder aber die Wiederverwendbarkeit des Contents durch die Trennung von Inhalt, Struktur und Gestaltung.<sup>34</sup> Diese Vorteile sind damit eng mit den Zielen dieser Arbeit zur Konzeption eines Softwaresystems zur internen Kommunikation verbunden. Folglich ist ein Content-Management-Systemen zur Datenpflege und –aufbereitung optimal geeignet, da es sich dabei, wie bereits erwähnt, um ein IT-System mit Content Related Technologien handelt. Den Vorteilen gegenüberstehend sind natürlich auch wirtschaftliche Überlegungen nicht zu vernachlässigen. Die Einführung eines Content-Management-Systems ist meist mit Lizenzkosten verbunden. Weiterhin kann die Implementierung mit Migrationskosten gekoppelt sein. Im speziellen Fall müssen die in den PowerPoint Folien gespeicherten Inhalte, unter Berücksichtigung des Touchscreen Konzepts, in das CMS überführt werden. Bei der Übertragung der Inhalte aus den PowerPoint Folien, ist es wie beschrieben erforderlich, diese zu strukturieren. Ferner fallen bei der Verwendung eines Content-Management-Systems ggf. Implementierungskosten, Einführungskosten sowie Administrations- und Wartungskosten an.<sup>35</sup>

Die vorgestellten Technologien des Content Managements lassen sich zur Konzeption eines IT-Systems zur internen Kommunikation verwenden. Aus diesem Grund wird ein Open Source Content Management Produkt zur Datenhaltung Anwendung finden. In dem folgenden Kapitel wird ein solches System, mit dem besonderen Focus der Strukturierungsmöglichkeiten, vorgestellt.

## 4.2 Enterprise-Content-Management – Alfresco Lösung

Das Management-System welches zur Contentverwaltung der Abteilung UH/I verwendet werden soll, ist das System von Alfresco. Dieses System bietet eine Vielzahl von Möglichkeiten und gehört laut Hersteller zur führenden Open Source Alternative im Bereich des Content Managements.<sup>36</sup> Bei der Alfresco Lösung handelt es sich um ein Enterprise-Content-Management-System. Diese Managementlösung unterstützt verschiedenste Con-

---

<sup>33</sup> vgl. Schmitz 2007, S. 204

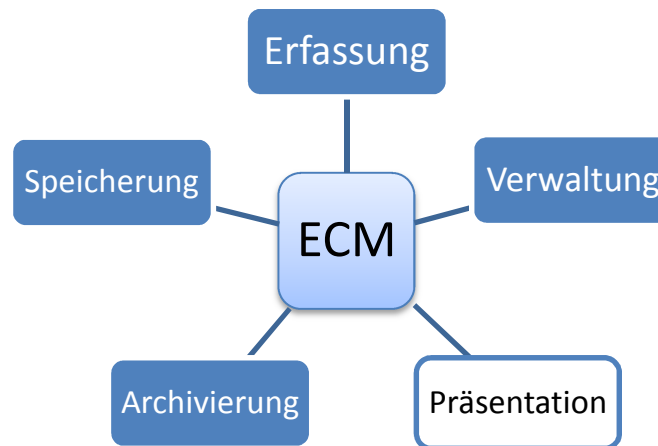
<sup>34</sup> vgl. Schmitz 2007, S. 204

<sup>35</sup> vgl. Schmitz 2007, S. 208

<sup>36</sup> vgl. Alfresco Software 2011 – Alfresco Produkte

tent Arten wie Dokumente, Bilder, Webinhalte oder XML Dokumente und ermöglicht die Verwaltung der Inhalte in strukturierter oder unstrukturierter Form.<sup>37</sup> Das Enterprise-Content-Management verwendet analog zum Content-Management-System die bereits vorgestellten Technologien, jedoch mit dem weiteren Anspruch, alle bestehenden CRT-Systeme zu integrieren.<sup>38</sup> Die grundlegende Systemarchitektur ist auf Basis der Association for Information and Image Management (AIIM) entwickelt. In diesem Zusammenhang sind Kernfunktionen eines Enterprise-Content-Management Systems abgebildet.

*Abbildung 5 – Komponenten des ECM*



*Quelle: in Anlehnung an: Riggert 2009, S. 6*

In Abbildung 5 sind die fünf Komponenten des Enterprise-Content-Managements zusammenfassend dargestellt. Die Komponente zur Präsentation der Inhalte ist in der Grafik hervorgehoben. Der Grund hierfür ist, dass im vorliegenden Fall die Ausgabe und Präsentation nicht durch das ECM System, sondern durch ein eigenentwickeltes System erfolgen soll. In den folgenden Unterkapiteln wird deshalb die Erfassung des Contents in den Vordergrund gestellt. In diesem Zusammenhang werden bereits bestehende inhaltliche Strukturen betrachtet, um die Basis für eine mögliche Migration und automatisierte Verarbeitung zu schaffen. Ausgangslage für diese Betrachtungen ist die von der Arbeitsgruppe Organisation und Kommunikation erarbeitete PowerPoint Präsentation. Die vollständige Präsentation ist der Arbeit im Anhang beigelegt.

---

<sup>37</sup> vgl. Caruana 2010, S.3

<sup>38</sup> vgl. Schmitz 2007, S. 211

## 4.2.1 Analyse der bestehenden Informationen

Wie bereits erwähnt wird aufgrund der bestehenden Vorteile das Enterprise-Content-Management-System als Datenspeicher für die Informationen zur internen Kommunikation verwendet. Im Folgenden wird eine Präsentationsfolie vorgestellt und deren Inhalt bewertet. Dies wird zum einen als Grundlage für die Definition verschiedener Content Arten genutzt. Auf der anderen Seite dient sie zur Bewertung der Inhalte, in wie weit eine Strukturierung zur automatischen Verarbeitung erforderlich ist.

Abbildung 6 – Präsentationsfolie; „Blick über den Tellerrand“

The slide is a presentation slide from Lufthansa Technik Logistik. It has a blue header bar with the text 'Information Technology HAM UH/I' and the Lufthansa Technik Logistik logo. Below the header is a grey bar with the title 'Blick über den Tellerrand'. The main content area is white and contains the following elements:

- Title: **Lufthansa stellt ein: 4.000 neue Mitarbeiter in Deutschland**
- Date: 06.01.11
- Image: A photograph of three Lufthansa employees (two women and one man) in uniform.
- Text: Lufthansa wird 2011 rund 4.000 neue Mitarbeiter in Deutschland einstellen. 2.200 Flugbegleiter/innen, 900 neue Servicemitarbeiter und 270 Pilotenschüler. Mit dem Airbus A380 wächst der Personalbedarf im Cockpit, in der Kabine und am Boden. Lufthansa zählt zu den attraktivsten Arbeitgebern Deutschlands, rund 115.000 Bewerbungen wurden im vergangenen Jahr über das Karriereportal abgegeben.
- Footer: LHT/red

Quelle: Lufthansa Technik Logistik GmbH – UH/I

Wie in Abbildung 6 erkennbar, ist der Content durch Texte, Bilder oder Grafiken definiert. In der blauen Kopfzeile einer jeden Folie ist der Titel der Abteilung (1), dessen Abkürzung (2) sowie auf der rechten Seite das Firmenlogo (3) zu sehen. Darunter ist in dem grauen Steifen das aktuelle Thema (4) beschrieben. Hierbei werden mehrere Folien einem gleichen Thema zugeordnet. In der weißen Fläche sind wiederum Informationen, die zwar bei jeder Folie inhaltlich verschieden, für Folien des gleichen Themas jedoch identisch angeordnet und formatiert sind. In der oben dargestellten Präsentationsfolie handelt es sich um den Titel (5) mit Untertitel (6), einem Bild (7), einem Artikel (8) sowie die Redaktion (9).

Dies zeigt, dass die verschiedenen Content Arten klar strukturiert sind. Selbst ohne die gegebenen Hinweise wären Überschriften, Untertitel oder Beschreibungen erkennbar. Ungeachtet dessen, ist die Struktur wie bereits im Kapitel 4.1.1 beschrieben, ausschließ-

lich über die Formatierung definiert. Wird demzufolge eine neue Folie erstellt so muss die Formatierung manuell an den durch die anderen Folien gegebenen Standard angepasst werden. Hierbei besteht ständig die Gefahr, dass Verletzungen der Corporate Identity auftreten oder schlicht falsche Formatierungen gewählt werden und das einheitliche Gesamtbild zerstören.

Aus diesem Grund wird für das vorliegende Softwarevorhaben eine eindeutige Struktur entwickelt, sodass anschließend die Formatierung und Darstellung automatisiert erfolgen kann. Hierzu ist es erforderlich, die Strukturiermechanismen der Alfresco Lösung zu betrachten. Es handelt sich um das so genannte *Content Modeling* und ist im folgenden Kapitel erläutert.

## 4.2.2 Content Modeling

Das *Content Repository* des Alfresco ECM ermöglicht zum einen das Speichern und Abrufen der Inhalte, entweder aus dem Dateisystem oder einer Datenbank. Ferner sind im Content Repository die Metadaten sowie die Content-Beziehungen verankert.<sup>39</sup> Ein wesentlicher Bestandteil des Content Repositories ist das Content Modeling. Content Modeling in Verbindung mit Alfresco bietet die Grundlage um die bereits erwähnte notwendige Strukturierung der Inhalte zu ermöglichen. Der logische Aufbau des Content Repositories ist als ein Baum<sup>40</sup> mit verschiedenen Knoten zu verstehen, wobei jeder Knoten eine oder mehrere Parameter besitzt. Die Parameter können verschiedenen Datentypen zugewiesen werden, welche einen oder auch mehrere Werte annehmen können. In diesem Zusammenhang werden mit Hilfe von Content Models folgende Restriktionen für die Datenstruktur vorgegeben.<sup>41</sup>

- der Typ eines Knoten wird festgelegt (Bsp. Verzeichnis oder Dokument)
- ein Knoten liefert eine bestimmte Anzahl an Parametern
- ein Parameter hat einen definierten Datentyp
- der Wert eines Parameters passt in ein festgelegtes Raster
- der Knoten muss in einer speziellen Beziehung zu einem anderen stehen

In Bezug auf das Alfresco System ist ein Content Model in einem XML Dokument definiert. Jedes Content Model ist wiederum durch ein so genanntes *Content Metamodel* be-

---

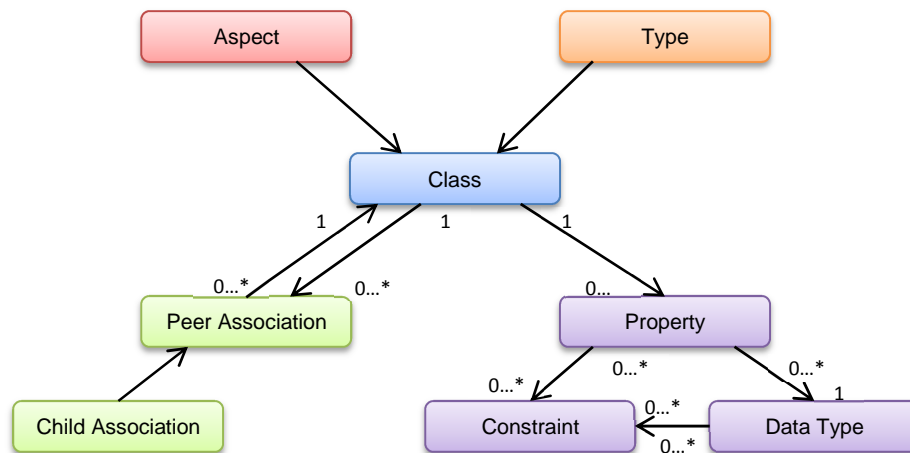
<sup>39</sup> vgl. Caruana 2010, S. 24

<sup>40</sup> Ein Baum ist eine Datenstruktur in der Informatik, deren Kenntnis vorausgesetzt wird.

<sup>41</sup> vgl. Caruana 2010, S. 75

schrieben und definiert das Schema für das in dem XML Dokument abgelegtem Content Model.

Abbildung 7 – Content Metamodel



Quelle: in Anlehnung an: Caruana 2010, S. 79

In der Abbildung 7 wird das Content Metamodel in Verbindung mit dem Content Repository aufgezeigt und definiert die individuellen Möglichkeiten zur Beschreibung von Inhalten. Die zentral abgebildete Klasse wird durch einen *Typen* beschrieben. Dieser Typ repräsentiert analog zur Objektorientierung ein definiertes Objekt aus der realen Welt inklusive der Vererbung der spezifischen Eigenschaften.<sup>42</sup> Wird bspw. eine neue Klasse vom Typ *Ordner* definiert, so erbt die Klasse bereits alle Eigenschaften der Klasse *Ordner*. In diesem Zusammenhang existieren bereits von Alfresco definierte Standardtypen, wie der Typ *Content*, der Analog der Klasse *object* in Java gesehen werden kann und Ausgangspunkt für alle weitere Klassen ist. Wird für eine neue Klasse ein oder mehrere *Aspects* definiert, so beschreiben diese auf ähnliche Weise wie der Typ die Vererbung von Eigenschaften. Der entscheidende Unterschied ist jedoch, dass *Aspects* eine mehrfache Vererbung von Eigenschaften und Beziehungen ermöglichen.<sup>43</sup> Die Beziehungen zwischen verschiedenen Typen werden im Content Metamodel durch *Associations* abgebildet. Hierbei wird zwischen *Child Association* und *Peer Association* unterschieden. Wird eine *Child Association* implementiert, so ist das Vatelement fest mit dem Kind verbunden. Operationen wie bspw. das Löschen des Vaters führen dazu, dass auch das Kind mitgelöscht wird. Die *Peer Associations* beschreiben eine Beziehungen zwischen zwei Objekten, in der keines dem anderen übergeordnet ist. Wie in der Abbildung oben hervorgeht, können bei der

<sup>42</sup> vgl. Caruana 2010, S. 79

<sup>43</sup> vgl. Caruana 2010, S. 79

Modellierung einer neuen Content-Klasse von null bis unendlich viele Beziehungen definiert werden. Abschließend werden noch die Eigenschaften einer Klasse betrachtet. Analog zu den Beziehungen besteht die Möglichkeit, unendlich viele oder auch keine Eigenschaften festzuhalten. Ist dies der Fall, besteht das Objekt ausschließlich aus den Eigenschaften des Typs oder der Aspects. Wie bereits kurz beschrieben, kann eine Klasse vom Typ Ordner mit zusätzlich Eigenschaften versehen werden, wie bspw. einem Feld für den Autor. Für diese Eigenschaften können spezielle Datentypen (Bspw. text, int, date, boolean) vorgegeben werden. Des Weiteren besteht die Möglichkeit die Eigenschaft auf bestimmte Eingabewerte zu beschränken.<sup>44</sup>

Zur Abbildung des Contents aus der vorgestellten PowerPoint Präsentation werden vorrangig verschiedene Ordner- und Dokumenttypen mit spezifischen Eigenschaften angelegt. In diesem Zusammenhang ist es möglich, den Content klar zu strukturieren. Um welche es sich dabei genau handelt, wird im Zuge des entwickelten Prototyps im Kapitel 6 beschrieben.

Wie bereits erwähnt, sind die Content-Strukturen Voraussetzung für die automatisierte Verarbeitung. Im vorliegenden Fall, speziell mit dem Hintergrund den Content in einer Java Applikation zu verarbeiten, werden im Folgenden die Content Management Interoperability Services betrachtet. Diese wiederum sind notwendig, um den strukturierten Content über die Java Applikation abzurufen.

### 4.2.3 Content Management Interoperability Services

Das Unternehmen Alfresco hat zusammen mit weiteren Firmen der Informationstechnologie im September 2008 den *Content Management Interoperability Services* (CMIS) Standard etabliert. Dieser Standard ermöglicht den Zugriff auf unterschiedliche Content-Management-Systeme auf eine einheitliche und interoperable Weise, sofern die Systeme den Standard unterstützen. Das Alfresco EMC ist neben Microsoft SharePoint, IBM FileNet und Weiteren ein solches System. Folglich gewährt CMIS einer Applikation den Zugang zu den Inhalten verschiedener CMS ohne dass diese direkt im System integriert ist. In diesem Zusammenhang stellt CMIS eine auf SQL-basierende Abfragesprache bereit, mit deren Hilfe der Content oder dessen Metadaten abgerufen werden können. Zusammengefasst bietet CMIS verschiedene Services zur Verwaltung der Content Metadaten, Versionierung, Auswertung der Ordnerinhalte oder deren Beziehungen. Die wichtigs-

---

<sup>44</sup> vgl. Caruana 2010, S. 79-91

ten der verschiedenen Dienste, in denen sich die Content Management Interoperability Services unterteilen lassen, sind im Folgenden kurz vorgestellt.

Als erstes sind die *Repository Services* zu nennen, sie dienen zur Identifizierung von verfügbaren Repositories durch die angebundene Applikation. Zudem liefert dieser Dienst grundlegende Informationen über verschiedene Typen des Repositories analog eines Data Dictionary.

Ein weiterer wichtiger Dienst ermöglicht die Navigation innerhalb des Repositories und gewährt den Zugriff auf die vorgestellte Baumstruktur. Es handelt sich um die *Navigation services*, mit deren Hilfe zudem der Abruf der Knoten oberhalb (Vater) oder unterhalb (Kinder) eines bestimmten Objekts möglich ist.<sup>45</sup> Liefert bspw. eine Abfrage einen Ordner, so besteht durch die Navigation Services die Möglichkeit, den kompletten Inhalt (Kinder) abzurufen.

Anhand der *Object Services* werden die grundlegenden Funktionen zur Verwaltung der Objekte, wie Dokumente, Ordner, Rechte oder Beziehungen bereitgestellt. Dazu gehören die Erstellung, das Lesen, Aktualisieren oder Löschen eines Objekts. Darüber hinaus besteht bei Dokumenten die Möglichkeit, die in den Metadaten verankerten Eigenschaften oder Rechte abzurufen sowie über einen Datenstrom den Inhalt des Dokuments auszulesen oder zu schreiben. Die Object Services ermöglichen den Zugriff auf die Objekte durch die Angabe des Pfades oder der ObjectID.<sup>46</sup> Wie bereits erwähnt, können die Objekte zudem mittels einer SQL-Abfrage gelesen werden. Die Bereitstellung einer solchen Abfrage wird durch die *Discovery Services* ermöglicht.<sup>47</sup>

Neben diesen vorgestellten Diensten bietet der CMIS Standard noch eine Vielzahl Weiterer. Diese werden jedoch nicht weiter ausgeführt, da die erläuterten Dienste für die Anbindung des Prototyps an das Content-Management-System vorerst ausreichend sind.

Zusammenfassend wurden in den vorangegangenen Kapiteln die Grundlagen für ein Systemkonzept zur internen Kommunikation dargelegt. Die folgenden Abschnitte dieser Arbeit werden darauf aufbauend ein Systementwurf vorstellen und die Machbarkeit anhand eines Prototypen aufzeigen.

---

<sup>45</sup> vgl. Caruana 2010, S. 72

<sup>46</sup> vgl. Caruana 2010, S. 72

<sup>47</sup> vgl. Caruana 2010, S. 72

## 5. Systementwurf

Die systematische Untersuchung der an das Softwareprodukt gestellten Anforderungen sowohl aus fachlicher, als auch aus der technischen Betrachtungsrichtung zählt zu den wichtigsten Tätigkeiten während des Software-Entwicklungsprozesses.<sup>48</sup> Die Anforderungen an ein System zur internen Kommunikation der IT-Abteilung wurden von verschiedenen Instanzen formuliert und festgehalten. Die optische Darstellung, die zu vermittelnden Inhalte sowie die Gruppierung der Inhalte sind weitestgehend durch die Arbeitsgruppe Kommunikation und Organisation vorgegeben. Weitere Anforderungen sind durch Interviews oder Befragungen von Know-How-Trägern aufgenommen worden. Abschließende Anforderungen wurden basierend auf einer IST-Analyse des derzeitigen Systems und die damit verbundenen Pflegeprozesse der Informationen definiert. In den folgenden Abschnitten werden eingangs grundlegende Vorgehensweisen zur Anforderungsdefinition kurz erläutert und anschließend die an das System gestellten Anforderungen beschrieben. Abschließend werden diese in einem geeigneten Umsetzungsszenario, bezogen auf die vorherigen Erkenntnisse, eingebunden.

Nach der Betrachtung der Anforderungen wird das Softwarevorhaben auf die Verfahrenswesen der Softwareentwicklung projiziert. In diesem Zusammenhang wird eine passende Möglichkeit zur Modellierung des Softwareprojekts selektiert und an ausgewählten Beispielen durchgeführt.

### 5.1 Anforderungsmanagement

Anforderungen sind in ihrer Art sehr unterschiedlich. Die verbal von verschiedenen Personen formulierten Forderungen an das System sind dabei meist sehr unscharf und nicht ausreichend. Folglich ist es erforderlich, sie auf Vollständigkeit, Konsistenz, Eindeutigkeit und Durchführbarkeit zu analysieren. Die Anforderungen gelten als vollständig, wenn sie genügen, um die Ziele des Softwareprodukts, mit allen benötigten Daten und Funktionen zu erreichen.<sup>49</sup> Durch die Vielseitigkeit der Anforderungen, verbunden mit den individuel-

---

<sup>48</sup> vgl. Balzert 2001, S. 98

<sup>49</sup> vgl. Balzert 2001, S. 102f



len Ansprüchen verschiedener Personen, besteht die Gefahr der Widersprüchlichkeit. Folgende zwei Anforderungen sollen betrachtet werden:

1. *Alle Daten sollen im System vollständig gespeichert und archiviert werden*
2. *Das Softwareprodukt soll sehr wenig logischen Datenspeicher benötigen*

Diese beispielhaft gewählten Anforderungen sind widersprüchlich. Treten bei aller definierten Anforderung keine Widersprüche auf, so gelten sie als konsistent. Die zweite Beispielanforderung ist zudem nicht eindeutig. Die Aussage „sehr wenig“ kann von verschiedenen Personen unterschiedlich verstanden werden. Folglich sind die Anforderungen so zu definieren, dass sie keine verschiedenen Interpretationsmöglichkeiten bieten. Ist dies bei allen Anforderungen der Fall, so sind sie eindeutig formuliert. Die abschließende Beurteilung der Durchführbarkeit der Anforderungen bezieht sich auf die technische Umsetzung. Beispielsweise kann es technisch nicht möglich sein, eine Antwortzeit für eine Funktion innerhalb von drei Millisekunden zu realisieren. In diesem Zusammenhang kann die Entwicklung eines Prototyps für einen Funktionstest, verbunden mit den technischen Gegebenheiten Abhilfe schaffen.<sup>50</sup> Die Anforderungen an das IT-System zur internen Kommunikation teilen sich in funktionelle Anforderungen, welche sich aus fachlichen Hintergründen ergeben und in technische Anforderungen bezogen auf die Hardware.

### 5.1.1 Funktionelle Anforderungen

In diesem Abschnitt werden ausgewählte funktionelle Anforderungen an das Softwaresystem beschrieben. Kernanforderung an das Softwareprojekt ist die einfache und effiziente Erfassung und Pflege der Informationen. Dies bedeutet im Einzelnen, dass für die Applikation ein Konzept zu entwickeln ist, welches einerseits die Pflege der Informationen und auf der anderen Seite deren Darstellung beschreibt. Folglich muss die Applikation Funktionen zum Abruf der Informationen aus einem Datenpool bereitstellen und gleichermaßen entsprechende Methoden zur geeigneten Darstellung dieser Vorhalten. Als Umsetzungsszenario für das Softwareprodukt bedeutet dies, dass die Applikation lediglich den Abruf und die Darstellung der Informationen sowie die Interaktion mit dem Anwender kontrolliert und steuert. Für die Pflege der Informationen wird aufgrund der Vorstudie das Alfresco ECM Anwendung finden. Eine weitere Anforderung an die Software ist eine intelligente Archivierungsfunktion verbunden mit einer selektiven Bereitstellung von Informationen. Begründet ist diese Anforderung durch die heutige PowerPoint Präsentation, in der veraltete

---

<sup>50</sup> vgl. Balzert 2001, S. 102f

Nachrichten manuell gelöscht und durch neue ersetzt werden. In Folge dessen muss die zukünftige Applikation Funktionen anbieten, mit deren Hilfe genau die Informationen abgerufen werden können, die in einem definierten zeitlichen Rahmen liegen. Das System als Datenspeicher wiederum, muss eine Art Metadatenverwaltung bereitstellen, um festzuhalten wann die Informationen gepflegt wurden. Eine Anforderung der Arbeitsgruppe Organisation und Kommunikation ist die einheitliche Darstellung der Informationen. In der heutigen Präsentation werden die Inhalte gepflegt und zeitgleich die optische Darstellung nach dem Corporate Design der Lufthansa Technik Logistik GmbH angepasst. Die Applikation soll angepasste Designs für die verschiedenen Kategorien vorhalten, um bspw. alle News identisch darzustellen. Eine letzte Anforderung, welche in diesem Rahmen betrachtet werden soll, ist die Umsetzung eines geeigneten Exception-Handlings. Das bedeutet, dass die Applikation in der Lage sein soll, bei Fehlverhalten jedweder Art den Betrieb fortzuführen oder wenn dies ausgeschlossen, ist den Anwender zu informieren, dass ein administrativer Eingriff durch den Applikationsverantwortlichen notwendig ist.

### **5.1.2 Technische Anforderungen**

Die technischen Anforderungen an das Softwaresystem ergeben sich aus den Hardwareeigenschaften des Touchscreenmonitors und der IT-Infrastruktur der Lufthansa Technik Logistik GmbH.

Bei dem genannten Touchscreenmonitor handelt es sich um einen Samsung LCD Monitor MG32PS mit integriertem Rechnermodul. Die Benutzerinteraktion mit einem Finger wird von dem Monitor erfasst und durch eine entsprechend installierte Systemsoftware in einen Mausklick umgesetzt. Für eine zu entwickelnde Applikation bedeutet dies, dass sie sich mit einer herkömmlichen Maus bedienen lassen muss. Die Systemsoftware unterscheidet nur über Umwege, ob es sich um einen Doppelklick, Rechts- oder Linksklick handelt. Das resultiert daraus, dass sich eine Fingerbedienung nicht eins zu eins auf eine Mausbedienung übertragen lässt. Die konkrete Anforderung an das Softwaresystem zur internen Kommunikation ist folglich eine Steuerung mit ausschließlich der linken Maustaste.

Aufgrund der eingeschränkten Rechen- und Speicherkapazität des im Monitor integrierten Rechnermoduls ist eine weitere technische Anforderung definiert. Der physische Speicherort der zur internen Kommunikation dargestellten Inhalte soll sich außerhalb des genannten Rechners befinden. Das Rechnermodul verfügt über einen Netzwerkanschluss und somit Zugang zum internen Firmennetzwerk über das die Informationen abgerufen werden können.

## 5.2 Systemmodellierung

Für die Systemmodellierung existieren in der modernen Softwareentwicklung verschiedene Betrachtungsweisen und Grundzusammenhänge. Es handelt sich um die Betrachtung des Systems aus der Sicht der Daten, Funktion, Dynamik oder Benutzeroberfläche. Zur Modellierung eines neuen Softwareprodukts ist es dabei meist nicht erforderlich, jede der vier Sichtweisen zu modellieren, da das Gewicht der einzelnen Sichten auf das System unterschiedlich sein kann. Diesbezüglich lassen sich Softwareprodukte in sechs Komplexitätsklassen unterteilen, in denen bestimmte Softwareeigenschaften dominieren. Es handelt sich um die Dimensionen: *Komplexität der Funktionen*, *Komplexität der Daten*, *Komplexität der Algorithmen*, *Komplexität des zeitabhängigen Verhaltens*, *Komplexität der Systemumgebung* oder die *Komplexität der Benutzeroberfläche*. Für jede dieser Komplexitätsklassen muss zur Modellierung ein passendes Konzept gewählt werden, um bspw. bei einem System welches eine Vielzahl an Funktionen bereitstellen muss, jede dieser Funktionen geeignet zu modellieren. Als mögliches Konzept könnte z.B. ein Funktionsbaum verwendet werden. Auf der anderen Seite ist bei einem Softwareprodukt mit dem Fokus auf der Datenhaltung, welches viele komplexe Datenstrukturen enthält, ein Entity Relationship Model zu empfehlen. Bei den angesprochenen Beispielen Funktionsbaum und Entity Relationship Model handelt es sich um Basiskonzepte, welche in den vergangenen Jahren entwickelt wurden.<sup>51</sup> Folgende Grafik stellt diese komplexen Zusammenhänge der Softwaremodellierung vereinfacht dar. Sie zeigt eine Übersicht der Anwendungsbereiche, der verschiedenen Komplexitätsklassen sowie der daraus resultierenden Basiskonzepte.

---

<sup>51</sup> vgl. Balzert 2001, S. 104f

Abbildung 8 – Abbildung von Anwendungsbereichen auf Basiskonzepte

Anwendungsbereich	Komplexität ...	Basiskonzepte
<div style="display: flex; flex-direction: column; align-items: center;"> <div style="margin-bottom: 10px;">↑ technisch- wissenschaftlich ↓</div> <div style="margin-bottom: 10px;">↑ administrativ ↓</div> <div style="margin-bottom: 10px;">↑ Echtzeit ↓</div> <div style="margin-bottom: 10px;">↑ Mensch- Computer- Interaktion ↓</div> </div>	...der Daten	<ul style="list-style-type: none"> <li>■ Data Dictionary</li> <li>■ ER (Entity Relationship)</li> <li>■ (Zustandsautomat)</li> </ul>
	...der Funktionen	<ul style="list-style-type: none"> <li>■ Klassendiagramm</li> <li>■ Geschäftsprozesse</li> <li>■ Datenflussdiagramm</li> <li>■ Funktionsbaum</li> </ul>
	...der Algorithmen	<ul style="list-style-type: none"> <li>■ Pseudocode</li> <li>■ PAP (Programmablaufplan)</li> <li>■ Struktogramm</li> <li>■ ET (Entscheidungstabelle)</li> <li>■ Regeln</li> </ul>
	...der Systemumgebung	<ul style="list-style-type: none"> <li>■ Datenflussdiagramm</li> </ul>
	...des zeitabhängigen Verhaltens	<ul style="list-style-type: none"> <li>■ Petri-Netz</li> <li>■ Zustandsautomat</li> <li>■ Aktivitätsdiagramm</li> <li>■ Sequenzdiagramm</li> <li>■ Kollaborationsdiagramm</li> </ul>
	...der Benutzungsoberfläche	<ul style="list-style-type: none"> <li>■ grafische Spezifikation (Petri-Netz)</li> <li>■ (Zustandsautomat)</li> <li>■ Kontrollstrukturen</li> <li>■ Regeln</li> </ul>

Legende: ( ) = eingeschränkt

Quelle: Balzert 2001, S. 107

Aufgrund der Vielzahl der verfügbaren Basiskonzepte lassen sich diese kombinieren und ergeben drei verschiedene Konzepte zur Systemanalyse. Es handelt sich um die *Echtzeit Analyse*, *Strukturierte Analyse* und *Objektorientierte Analyse (OOA)*.<sup>52</sup> Im Folgenden wird die Abbildung 8 auf das Softwaresystem zur internen Kommunikation angewendet. Die Software für den Touchscreenmonitor fällt in den technisch/wissenschaftlichen Bereich mit einer sehr hohen Komplexität der Funktionen. Die in diesen Bereichen anzuwendenden Basiskonzepte sind durch die Objektorientierte Analyse vereint.<sup>53</sup>

## 5.2.1 Produkteinsatz

Der Anwendungsbereich der Applikation, welche den Projektnamen L/information tragen soll, ist der IT Bereich der Lufthansa Technik Logistik. Daraus ergeben sich die Zielgruppen, welche sich von den Mitarbeitern der Abteilung über interne und externe Besucher oder Dienstleister bis hin zu internen oder externen Kunden erstrecken. Die Betriebsbedingungen sind die Räumlichkeiten der LTL.

<sup>52</sup> vgl. Balzert 2001, S. 106

<sup>53</sup> vgl. Balzert 2001, Abb. 2.2-4, S. 108

Zur geeigneten Anwendung von L/information wurden von der Arbeitsgruppe Organisation und Kommunikation Kategorien entwickelt um die verschiedenen Zielgruppen anzusprechen. Es handelt sich um die fünf Kategorien: Applikationen, Lufthansa Konzernstruktur, News, Abteilung UH/I und Projekte. Alle im Softwaresystem verfügbaren Inhalte sind einer dieser Kategorien zuzuordnen. In diesem Zusammenhang hat der Anwender die Möglichkeit, mit Hilfe der Touchfähigkeit des Monitors gezielt in den einzelnen Rubriken zu navigieren. Das durch die Arbeitsgruppe entworfene Touchscreen Konzept sieht ferner für bestimmte Bereiche Unterkategorien vor. Diesen Unterkategorien, wie bspw. die einzelnen Unternehmen innerhalb der Lufthansa Konzernstruktur, können analog zu den im Folgenden beschriebenen Hauptkategorien modelliert werden und sind daher nicht Gegenstand dieser Arbeit.

## 5.2.2 Systembeschreibung

Das Softwaresystem wird aufgrund des Einsatzes auf dem Touchscreenmonitor als Java Desktopapplikation entwickelt. Des Weiteren wird im Systemverbund das Alfresco ECM integriert, welches als Datenspeicher fungiert. Folglich ergeben sich für die Applikation zwei verschiedene Akteure. Ein Akteur versteht sich als jemand, der aktiv und unmittelbar mit dem Softwaresystem interagiert. In diesem Zusammenhang kann ein Akteur eine Person oder auch eine Personengruppe sein, die mit der Software kommuniziert.<sup>54</sup> Im Fall dieses Softwarevorhabens handelt es sich zum einen um den Nutzer der Java Applikation, welcher diese dazu benutzt, um gezielt Informationen abzurufen. Zum anderen der Nutzer des Content-Management-Systems, der die relevanten Informationen pflegt und aufbereitet. Um die Akteure für die folgenden Betrachtungen klar zu definieren wird der erste Akteur als Kunde und der zweite als Mitarbeiter bezeichnet. In der Literatur ist es weiterhin üblich, extern beteiligte Systeme ebenfalls als Akteur zu kennzeichnen, welcher mit dem System kommuniziert.<sup>55</sup> Im vorliegenden Fall wäre das Alfresco Enterprise-Content-Management-System ein solches externes System. Zur Umsetzung der definierten Ziele sind die Schnittstellen zwischen Alfresco und der Java Applikation zu beschreiben. Ferner ist eine Struktur innerhalb des Enterprise-Content-Management-Systems zu konzipieren, die die Differenzierung der zur Verfügung gestellten Daten ermöglicht. Im Folgenden wird deshalb ein Gesamtsystem bestehend aus beiden Applikationen betrachtet. Nachstehende Grafik zeigt eine erste grobe Produktübersicht. Dieses so genannte

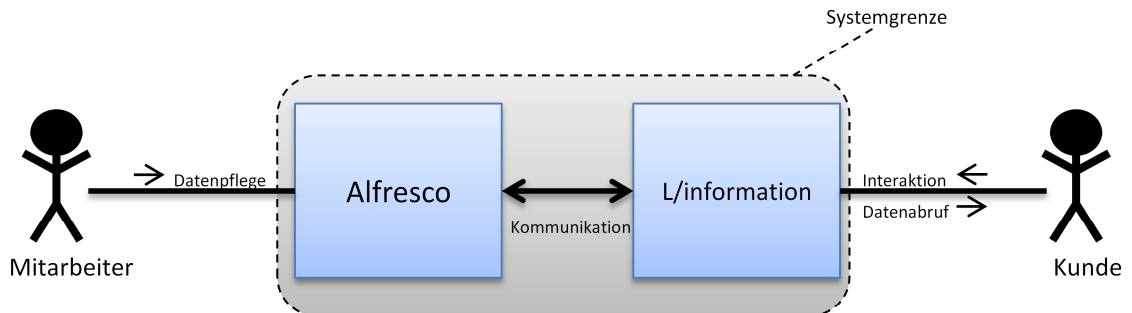
---

<sup>54</sup> vgl. Balzert 2001, S. 64

<sup>55</sup> vgl. Balzert 2001, S. 126f

Umweltdiagramm zeigt den Einfluss der Akteure auf das System, wobei die inneren Systemabläufe noch nicht näher beschrieben sind.<sup>56</sup>

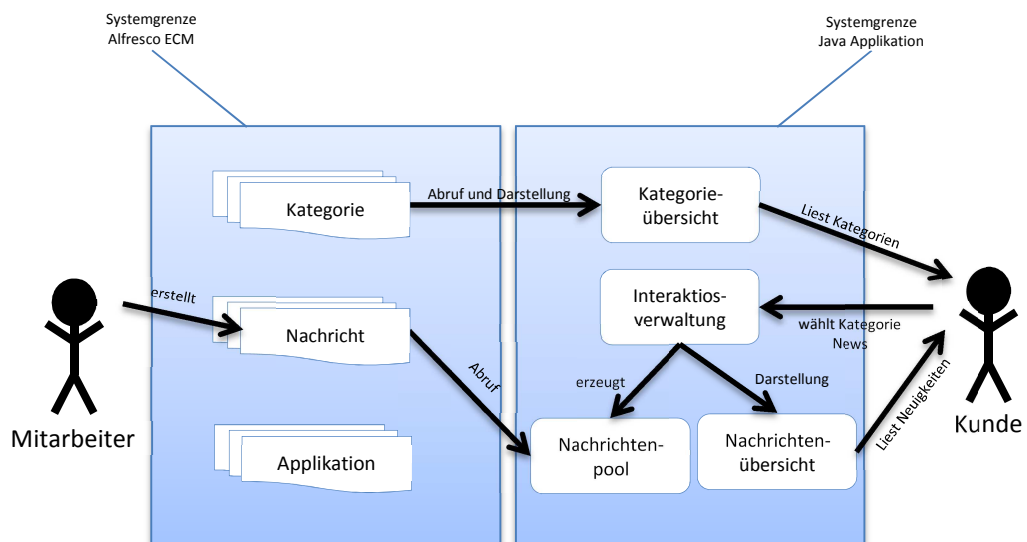
Abbildung 9 – Umweltdiagramm



Quelle: Eigene Abbildung

Um die Zusammenarbeit des Alfresco Enterprise-Content-Management-Systems mit L/information besser zu verstehen, wurde folgender Grobentwurf entwickelt. Dieser zeigt am Beispiel der Pflege und des Abrufes der News die grundsätzliche Funktionsweise der gesamten Applikation.

Abbildung 10 – Grobentwurf des Systems



Quelle: Eigene Abbildung

<sup>56</sup> vgl. Balzert 2001, S. 64f

Wie in Abbildung 10 erkennbar, sind in Alfresco mehrere Kategorien abgelegt. Das Java Programm lädt diese aus dem Enterprise-Content-Management-System und zeigt sie dem Kunden an. Dieser wählt eine Kategorie aus und selektiert diese. Im Beispiel wählt der Akteur Kunde die News. Das Java Programm reagiert auf die Benutzereingabe und erzeugt einen Nachrichtenpool. Dieser wird aus einzelnen Nachrichten, die aus dem Content-Management-System abgerufen werden zusammengestellt. Die separaten Nachrichten wiederum werden vom Akteur Mitarbeiter in Alfresco gepflegt. Auf Basis des Nachrichtenpools, werden die Neuigkeiten dem Kunden angezeigt. Durch die Zweiteilung des Systems verbunden mit der Datenspeicherung im Alfresco Enterprise-Content-Management-Systems und deren Darstellung durch die Java Applikation lässt sich die eingangs erwähnte Kernanforderung realisieren. Die in Abbildung 10 dargestellte Grobübersicht hat nicht den Anspruch der Vollständigkeit bezüglich des Systemumfangs. Die wichtigsten Funktionen werden im Folgenden durch ausgewählte Use Case Diagramme für Alfresco und für L/information betrachtet und beschrieben. Der Fokus liegt dabei auf denen der Java Applikation. Ein Use Case oder auch Anwendungsfall definiert alle Aufgaben an das Softwaresystem, die notwendig sind, wenn ein Akteur ein bestimmtes Ziel erreichen möchte.<sup>57</sup>

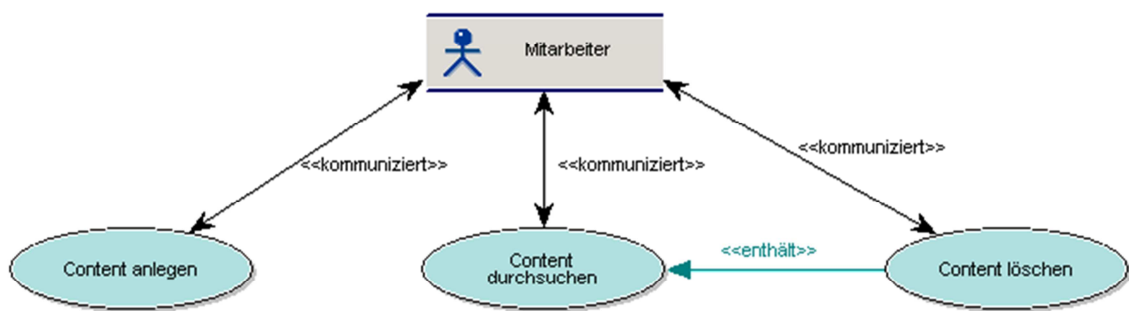
#### **5.2.2.1 Use Cases Alfresco**

Wie bereits erwähnt, fungiert Alfresco als Enterprise-Content-Management-System für die Gesamtapplikation als Datenspeicher. Die in diesem Zusammenhang verbundenen Standardfunktionalitäten, werden wie im Punkt 4.2 beschrieben, durch die Alfresco Lösung abgedeckt. Die daraus resultierenden Use Cases sind in folgendem Diagramm dargestellt. Weitere Use Cases wie Beispielsweise die Einrichtung einer Nutzerverwaltung und damit verbundene verschiedene Sichtweisen auf den Content werden im Zuge dieser Arbeit nicht betrachtet. Dieser Gesichtspunkt wird im Kapitel 6.3 aufgegriffen in welchem weitere Hinweise zum Weiterentwicklungspotential des vorgestellten Konzepts vorgestellt werden.

---

<sup>57</sup> vgl. Balzert 2001, S. 64f

Abbildung 11 – Übersichtsdiagramm Alfresco



Quelle: Eigene Abbildung – erzeugt mit objectiF 7.1

Für die in Abbildung 11 dargestellten Use Cases handelt es sich um abstrahierte Anwendungsfälle zur Pflege von Informationen (Content). Im speziellen Fall kann es sich bspw. um Neuigkeiten, Applikationen oder Mitarbeitersteckbriefe handeln. Die informationstechnische Herangehensweise ist dabei in allen Fällen gleich, deshalb wird in der folgenden Beschreibung nicht weiter zwischen verschiedenen Content-Arten unterschieden.

Der Anwendungsfall *Content anlegen* hat das Ziel neue Informationen vollständig im Content-Management-System abzuspeichern. Als Vorbedingung muss sich der Anwender am System anmelden. Wird der neue Content erfolgreich hochgeladen so steht er nach Abschluss des Anwendungsfalls zum Abruf über die Java Applikation bereit. Für den Fall, dass die neuen Informationen vom Mitarbeiter nicht vollständig abgespeichert werden, führt der Datenabruf durch L/information zu einem Fehler.

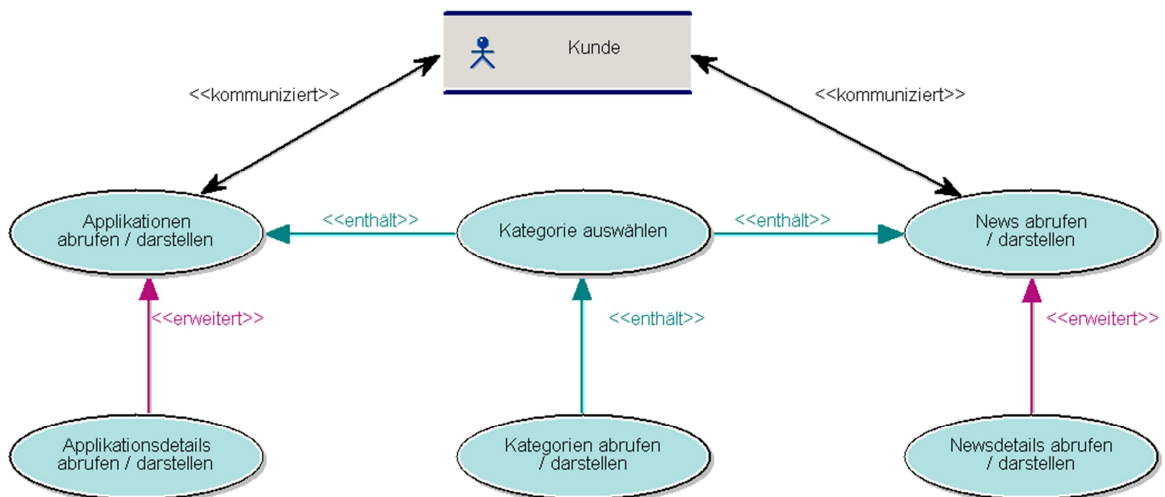
Das Ziel des Anwendungsfalls *Content löschen* ist die vollständige Beseitigung von abgelegten Informationen, so dass sie durch die Java Applikation nicht mehr zur Verfügung stehen. Ebenso wie beim Anlegen von neuen Informationen muss der Akteur erfolgreich am System angemeldet sein. Das Löschen eines beliebigen Contents setzt voraus, dass dieser vom Mitarbeiter im System gesucht wird. Folglich ist der Anwendungsfall *Content Durchsuchen* ein notwendiger Bestandteil zum Löschen von Informationen. Der Anwendungsfall *Content löschen* gilt als erfolgreich abgeschlossen, so dass alle notwendigen Informationen durch den Mitarbeiter entfernt wurden.



### 5.2.2.2 Use Cases L/information

Bei den in folgender Abbildung dargestellten Use Cases handelt es sich um die grundlegenden Anwendungsfälle zum Abruf von Informationen sowie deren Darstellung mittels L/information. Um die Komplexität der Grafik einfach zu halten, wurde das Übersichtsdiagramm auf den Abruf von Applikationen und News beschränkt.

Abbildung 12 – Übersichtsdiagramm L/information



Quelle: Eigene Abbildung – erzeugt mit objectiF 7.1

Auslösendes Ereignis des Anwendungsfalls *Applikationen abrufen/darstellen* besteht darin, dass sich der Kunde alle Applikationen der LTL in einer Übersicht auf dem Monitor darstellen lassen möchte. Folglich gilt für diesen Use Case die Vorbedingung, dass L/information erfolgreich gestartet und eine Verbindung zum Content-Management-System aufgebaut hat. Damit die Möglichkeit besteht die Applikationen darzustellen, muss der Akteur eine *Kategorie auswählen* – die der Applikationen. In diesem Zusammenhang muss das Softwaresystem alle verfügbaren Kategorien dem Anwender zur Auswahl zur Verfügung stellen. Um dies zu ermöglichen müssen alle Kategorien von dem Softwaresystem abgerufen und dargestellt werden. Das Abrufen der Kategorien ist folglich ebenso Bestandteil des Use Cases *Kategorie auswählen*. Der Anwendungsfall *Applikation abrufen* gilt als erfolgreich abgeschlossen, wenn die in Alfresco gepflegten Applikationen abgerufen und in einer Übersicht dem Akteur zur Verfügung gestellt werden.

Der Anwendungsfall *Applikationsdetails abrufen/darstellen* ist eine Erweiterung des Use Cases *Applikationen abrufen/darstellen*. Daraufhin gilt für diesen Anwendungsfall die Vor-

bedingung, dass alle im Content-Management-System abgelegt Applikationen abgerufen und dem Nutzer angezeigt werden.

### 5.2.3 Grobentwurf Systemarchitektur

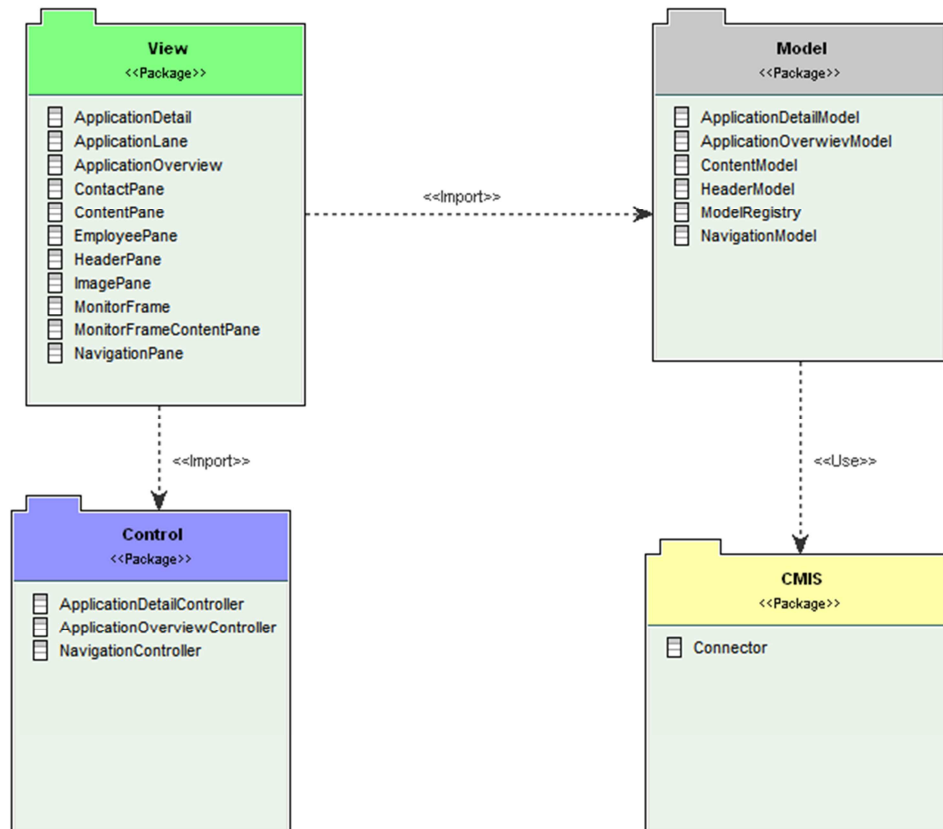
Aufgrund der Komplexität des Softwareproduktes und der damit verbundenen Vielzahl an Java Klassen werden diese auf einer höheren Ebene abstrahiert. Man spricht von Paketen (engl. packages). Die einzelnen Java Klassen umfassen einen bestimmten Themenbereich und gehören somit auch logisch zusammen. Es besteht ferner die Möglichkeit, im Zuge der Weiterentwicklung nicht nur einzelne Klassen, sondern auch ganze Pakete auszutauschen.<sup>58</sup> Aufgrund der grafischen Oberfläche der Applikation L/information wurden die Klassen zum einen nach dem *Model-View-Controller* Prinzip (MVC) gruppiert. Es handelt sich dabei um die Aufteilung der Klassen in die drei Basispakete *Modell*, *View* und *Controller*. Das Model fungiert hierbei als Datenspeicher und verarbeitet die gesamte Logik der Applikation, der View stellt die grafischen Komponenten der Oberfläche zur Verfügung. Zur Reaktion auf verschiedene Benutzerinteraktionen innerhalb der grafischen Oberfläche dient der Controller, welcher die Interaktionen erkennt und eine Reaktion darauf im dazugehörigen Model anstößt.<sup>59</sup> Neben diesen drei Basispaketen wurde zum anderen ein weiteres Paket definiert, in dem die Klassen zur Kommunikation mit dem Alfresco System verankert sind. Das Paketdiagramm stellt im Folgenden nicht nur die beschriebenen Pakete mit deren logische Verbindung dar, sondern wurde bereits um die wesentlichen Klassen innerhalb der Pakete erweitert.

---

<sup>58</sup> vgl. Balzert 2001, S. 205f

<sup>59</sup> vgl. Krüger 2008, S. 787f

Abbildung 13 – Grobentwurf Systemarchitektur



Quelle: Eigene Abbildung – erzeugt mit objectiF 7.1

Die Klassen innerhalb der Pakete sind analog der Kategorien getrennt. Zur Anzeige einer Hauptübersicht existieren sowohl ein View, ein dazugehöriges Model und ein Controller. Im Zuge der Umsetzung als Prototyp wurde der Fokus auf den Abruf und die Anzeige der Applikationen gelegt. In diesem Zusammenhang sind verschiedene Klassen zur Darstellung und Steuerung vorhanden. Das Paket CMIS enthält eine Connector-Klasse, mit der die Verbindung zum Enterprise-Content-Management System Alfresco aufgebaut wird, so dass die Informationen abgerufen werden können.

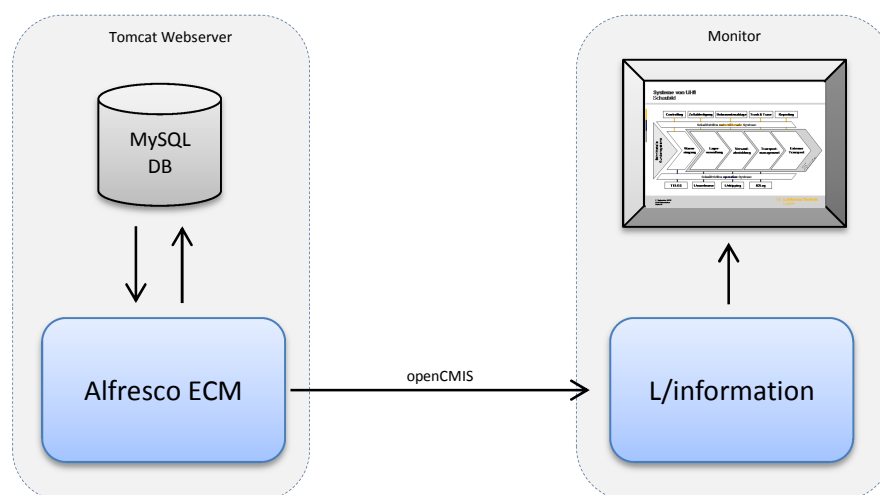
Anhand eines Prototypen wird aufgrund der gewonnenen Erkenntnisse im folgenden Abschnitt die Realisierungsmöglichkeit des entworfenen Systemkonzepts demonstriert. In diesem Zusammenhang wird zum einen eine Content-Strukturierung für Alfresco vorgeschlagen und die Realisierung dargestellt. Zum anderen wird der logische Aufbau der Java Applikation im Zuge des Datenabrufes beschrieben sowie die damit verbundenen Methoden erläutert.

## 6. Entwicklung des Prototyps

Das folgende Kapitel beschreibt auf Basis der bisherigen Erkenntnisse die konkrete Entwicklung eines Prototypen der die Machbarkeit des Softwarevorhabens demonstriert. Hierbei ist festzuhalten, dass nicht der Anspruch alle der an das Softwaresystem gestellten Anforderungen umzusetzen besteht.

Zur Migration der Inhalte in das Alfresco System wird zuerst das Content Modelling vorgestellt. In diesem Zusammenhang werden verschiedene definierte Ordnerstrukturen, sowie verwendete Metadaten zur automatisierten Identifikation, beschreiben. Anschließend wird der Datenabruf durch die Java Applikation, insbesondere der logische Umgang mit den vorgestellten Content-Strukturen dargestellt. Darüber hinaus werden exemplarisch die Methoden zur Abfrage der Inhalte in Verbindung mit den Content-Management Interoperability Services abgebildet. Anschließend folgen Hinweise zur Weiterentwicklung ergänzt durch einen Ausblick, der aus der Anwendung des in dieser Arbeit entwickelten Konzepts resultiert. Durch die Anwendung des Alfresco Systems in Verbindung mit der Java Applikation L/information ergibt sich folgende Systemarchitektur für das Softwarevorhaben.

Abbildung 14 – Systemarchitektur



Quelle: Eigene Abbildung

Das Alfresco ECM wird auf einem Tomcat Webserver mit Zugang zu einer MySQL Datenbank zur physikalischen Speicherung der Inhalte integriert. Direkt auf dem Touchscreenmonitor wird die Java Applikation als Clientanwendung implementiert. Über das Firmen-

netzwerk der Lufthansa Technik Logistik GmbH werden anschließend die Inhalte mit Hilfe der openCMIS Client API aus dem Content Repository abgerufen.

## 6.1 Entwicklung der Content Struktur mithilfe von Alfresco

Im Kapitel 4 wurde bereits darauf eingegangen, wie wichtig es ist den Content zu strukturieren. Im Zuge des vorgestellten Content Modeling werden im Folgenden die Inhalte zur internen Kommunikation mittels Alfresco einer Struktur zugewiesen. Es wird dabei vordergründig auf die Ablage der Hauptkategorien, News sowie der Applikation eingegangen. Diese individuellen Content Models wurden in Verbindung mit der Analyse der derzeitigen Präsentation sowie auf Grundlage des Touchscreen Konzepts entwickelt. Wie bereits kurz erwähnt, werden zur Definition eigener Content Models verschiedene XML-Dateien verwendet. Die Semantik der Dateien sowie deren Zusammenwirken wird nachfolgend erläutert.

### 6.1.1 Kategorie Model

Für die Gruppierung der Inhalte zur internen Kommunikation wurden von der Arbeitsgruppe Kommunikation und Organisation Kategorien definiert. Zur Ablage dieser wurde ein so genanntes *custom content model* angelegt. Die folgende Abbildung zeigt einen Auszug aus dem Quelltext dieser XML-Datei.

Abbildung 15 – Quelltextauszug: *kategorieModel.xml*

```
<namespaces>
  <!-- Define a Namespace for my new definitions -->
  <namespace uri="custom.kategorie.model" prefix="kt"/>
</namespaces>

<types>
  <type name = "kt:folder">
    <title>Kategorie Ordner</title>
    <parent>cm:folder</parent>
    <properties>
      <property name="kt:subtitle">
        <title>Beschreibung der Kategorie</title>
        <type>d:text</type>
        <mandatory>true</mandatory>
      </property>
    </properties>
  </type>
</types>
</model>
```

In der Abbildung ist zu erkennen, dass für ein neues Model ein Namensraum angegeben wird. Dieser Namensraum ermöglicht die eindeutige Spezifizierung eines Content Models mit Hilfe eines Namens und darf aus diesem Grund in allen Models nur einmal vorkommen. Der Namensraum besteht aus zwei Komponenten – einer URI und einem Prefix. Anhand des Prefix werden im weiteren Verlauf alle Parameter versehen und sind somit mit dem Namensraum verbunden. Aus diesem Grund ist ein Prefix zu wählen, dessen Interpretation auf die URI und somit auf den Content schließen lässt. Im vorliegenden Fall wurde die URI `"custom.kategorie.model"` und das Prefix `"kt"` gewählt. Weiterhin ist es mit Hilfe der URI des Namensraumes möglich, bereits bestehende Models zu importieren und auf diese innerhalb des Models zuzugreifen.

Neben dem Namensraum können für jedes Model ein oder mehrere Typen angelegt werden. Dies ist in der zweiten Sektion der Abbildung 15 erkennbar. Der definierte Typ besteht aus einem eindeutigen Namen. Anschließend werden für diesen Typ ein Titel, Vaterknoten sowie die Eigenschaften beschrieben. Für das Kategorie Model wurde nur ein Typ angelegt – `"kt:folder"`. Es handelt sich um einen Typen mit dem Titel „Kategorie Ordner“. Als Vaterknoten wurde `"cm:folder"` registriert. Das Prefix „cm“ ist im Zusammenhang mit der genannten Importmöglichkeit verbunden und beruft sich auf das allgemeine Content Model von Alfresco. Folglich erbt der neu definierte Typ alle Parameter und Eigenschaften eines Standardordners, wie er von Alfresco im Content Model definiert wurde. Zusätzlich zu diesen, wurde eine weitere Eigenschaft definiert, welche in dem Abschnitt Properties der XML-Datei zu finden ist. Zu sehen ist die spezielle Eigenschaft für eine Kategorie Beschreibung und trägt den Namen `"kt:subtitle"`. Zusätzlich wurde für diese Eigenschaft noch der Typ festgelegt und bestimmt, dass es sich um ein Pflichtfeld handelt.

Zusammenfassend wurde im Kategorie Model ein neuer Ordner beschrieben, welcher alle Eigenschaften eines Alfresco Standardordners besitzt und außerdem um die Angabe einer Beschreibung ergänzt wird. Die Erzeugung dieser XML-Datei versetzt Alfresco jedoch nicht in die Lage, diesen individuellen Content zu verwalten. Es ist zudem erforderlich, dieses neue Model im Content Repository zu registrieren. Dies geschieht im so genannten *Dictionary Bootstrap*, welches ebenso als XML-Datei definiert wird. Die folgende Abbildung zeigt die Registrierung des Content Models.

Abbildung 16 – Quelltextauszug: custom-model-context.xml

```
<!-- Registration of new models -->
<bean id="extension.dictionaryBootstrap" parent="dictionaryModelBootstrap" depends-
on="dictionaryBootstrap">
    <property name="models">
        <list>
            <value>alfresco/extension/kategorieModel.xml</value>
            <value>alfresco/extension/newsModel.xml</value>
            <value>alfresco/extension/employeeModel.xml</value>
            <value>alfresco/extension/applicationModel.xml</value>
        </list>
    </property>
</bean>
```

Die Abbildung 16 zeigt alle registrierten individuellen Models für das Content Repository. Dabei wird die XML-Datei, welche das Model beschreibt in einer Liste mit Angabe des Pfades aufgeführt. Das zuvor beschriebene Kategorie Model ist als erster Eintrag dieser Liste vorzufinden. Alle weiteren Models wurden analog dem Kategorie Model eingepflegt. Nachdem das Model im Content Repository angelegt wurde, besteht die Möglichkeit einen Kategorie Ordner mit den vorgestellten Eigenschaften anzulegen. Damit dies auch über die Alfresco Weboberfläche möglich ist, muss der Web Client von Alfresco entsprechend konfiguriert werden. Dies geschieht ebenfalls in einer XML-Datei Namens web-client-config und ist in folgender Abbildung vorgestellt.

Abbildung 17 – Quelltextauszug: web-client-config-custom.xml

```
<config evaluator="string-compare" condition="Space Wizards">
    <folder-types>
        <type name="kt:folder" icon="/images/icons/folder.gif" description= "Kategorie
Ordner" />
        <type name="news:folder" icon="/images/icons/folder.gif" description= "News Ord-
ner" />
        <type name="emp:folder" icon="/images/icons/folder.gif" description= "Mi-
tarbeiter" />
        <type name="appl:folder" icon="/images/icons/folder.gif" description= "Applika-
tion" />
    </folder-types>
</config>

<config evaluator="node-type" condition="kt:folder">
    <property-sheet>
        <show-property name="kt:subtitle" display-label="Beschreibung"/>
    </property-sheet>
</config>
```

Die Konfiguration der Alfresco Weboberfläche unterteilt sich in zwei Gruppen, welche auch in Abbildung 17 erkennbar sind. Die erste ist erforderlich, um bei der Anlage eines neuen Ordners, die individuell definierten Ordner auswählen zu können. Diesbezüglich

wird die Referenz auf den Typen gegeben, ein Symbol für den Ordner vergeben und eine Beschreibung für diesen festgelegt. Die zweite Gruppe wird dazu verwendet, um nach der Auswahl des individuellen Ordnerstyps, die definierten Eigenschaften über die Weboberfläche abzufragen. Für den vorgestellten Kategorie Ordner wurde folglich die Eigenschaft "kt:subtitle" mit dem Label "Beschreibung" in der XML-Datei eingetragen.

### 6.1.2 Application Model

Die Abteilung Informationstechnologie betreut und verwaltet verschiedene Applikationen über die Informationen intern zu kommunizieren sind. Aus diesem Grund, wurde auch für die spezielle Kategorie Applikationen ein Custom Model entworfen, dessen Quelltext im Folgenden vorgestellt und erläutert wird. Hierbei wird auf die Verfahrensweisen zur Registrierung des neuen Models im Content Repository, sowie die Konfiguration der Web-oberfläche nicht erneut eingegangen, da diese analog des vorgestellten Kategorie Models sind.

Abbildung 18 – Quelltextauszug: applicationModel.xml

```
<namespaces>
  <!-- Define a Namespace for my new definitions -->
  <namespace uri="custom.appl.model" prefix="appl" />
</namespaces>
<types>
  <type name = "appl:folder">
    <title>Applikation Ordner</title>
    <parent>cm:folder</parent>
    <properties>
      <property name="appl:appname">
        <title>Applikationsname</title>
        <type>d:text</type>
        <mandatory>true</mandatory>
      </property>

      <property name="appl:emp">
        <title>Applikationsverantwortung</title>
        <type>d:text</type>
        <mandatory>true</mandatory>
      </property>
    </properties>
  </type>
```



```

<type name = "appl:document">
  <title>Applikation</title>
  <parent>cm:content</parent>
  <properties>
    <property name="appl:name">
      <title>Name</title>
      <type>d:text</type>
      <mandatory>true</mandatory>
    </property>

    <property name="appl:description">
      <title>Kurzbeschreibung</title>
      <type>d:text</type>
      <mandatory>true</mandatory>
    </property>

    <property name="appl:owner">
      <title>Redaktion</title>
      <type>d:text</type>
      <mandatory>false</mandatory>
    </property>

    <property name="appl:date">
      <title>Datum</title>
      <type>d:text</type>
      <mandatory>true</mandatory>
    </property>
  </properties>
</type>
</types>

```

Entsprechend dem Kategorie Model wurde für das Application Model ein Namensraum vergeben. Für das Model wurde die URI "custom.appl.model" mit dem Prefix "appl" gewählt. Ebenso vergleichbar mit dem Kategorie Model ist im Model ein Ordner definiert, welcher die Eigenschaften des Standardordners erbt und durch zusätzliche Eigenschaften ergänzt ist. Der Applikationsordner ist durch die Eigenschaften "appl:appname" und "appl:emp" erweitert, welche zur Hinterlegung des Namens der Applikation sowie des Applikationsverantwortlichen dienen. Das Feld für den Applikationsnamen wurde definiert, weil der Ordnername, der vom Väterelement geerbt wird, bestimmte Sonderzeichen nicht zulässt. Die Applikationsverantwortung eines Systems bei der LTL ist bestimmten Mitarbeitern zugeordnet. Durch die Angabe der Personalnummer eines Mitarbeiters in den Metadaten eines Applikationsordners, wird diese Zuordnung umgehend beim Anlegen einer neuen Applikation im Content-Management-System hergestellt.

Im Application Model wurde ein weiterer Typ definiert. Es handelt sich um den Typ "appl:document", welcher vom Alfresco Standarddokument abgeleitet ist. Die Spezialisierung dieses Dokuments geht aus den zusätzlichen Eigenschaften Applikationsname,

Kurzbeschreibung, Redaktion und Datum hervor. In der XML-Datei ist erkennbar, dass diese spezifischen Eigenschaften als Metadaten beschrieben sind, welche folglich beim Anlegen dieses Dokuments vom Typ "appl:document" angegeben werden müssen.

### 6.1.3 News Model

Ein weiteres Custom Model, welches im Zuge dieser Arbeit erstellt wurde, ist das News Model. Dieses Model wird dazu verwendet, um Neuigkeiten der Abteilung UH/I oder auch allgemeine Neuigkeiten im Content-Management-System abzulegen. Hierzu wird im News Model vergleichbar dem Application Model ein Ordner sowie ein News Dokument definiert. Beide sind durch spezifische Eigenschaften in Verbindung mit Neuigkeiten beschrieben. Hierbei handelt es sich bspw. um die Redaktion, das Datum oder das Thema, zu dem die Nachricht zugeordnet werden kann. Eine Besonderheit beim News Model ist ein spezieller Typ, der auf das eben beschriebene News Dokument zurückgreift. Es handelt sich um ein Bild, für das keine speziellen Eigenschaften angegeben werden, wie die folgende Abbildung zeigt.

*Abbildung 19 – Quelltextauszug: newsModel.xml*

```
<type name = "news:picture">
  <title>NewsBild</title>
  <parent>news:document</parent>
  <properties>

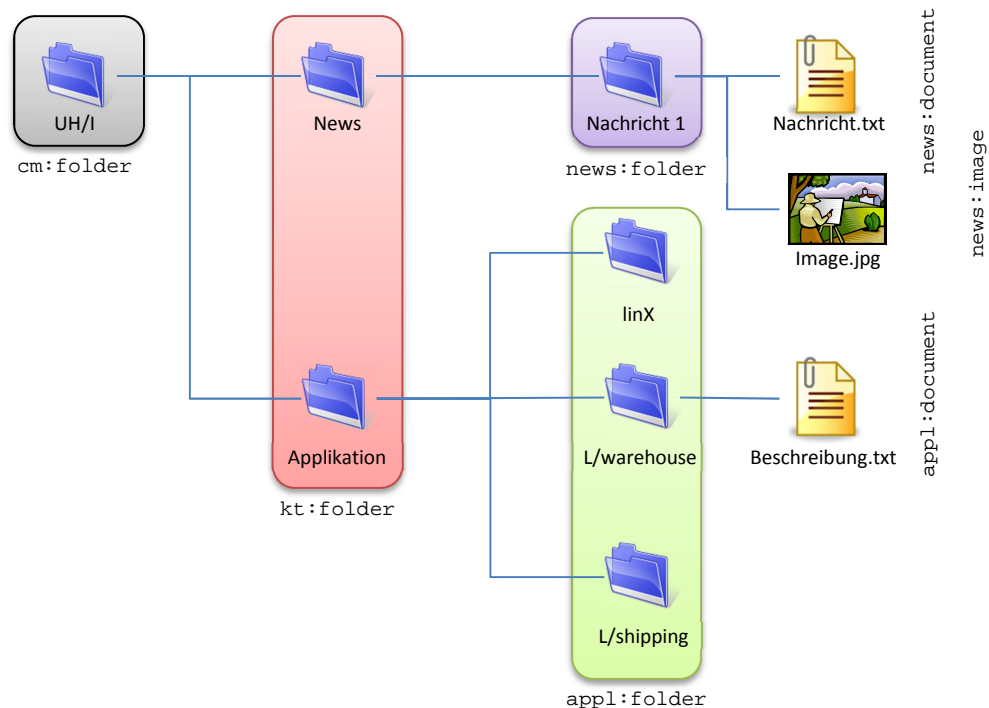
  </properties>
</type>
```

Die Anwendung dieses Typs erlaubt es trotz fehlender spezifischen Eigenschaften im Vergleich zum News Dokument, Bilder vom Text automatisiert zu differenzieren.

### 6.1.4 Strukturkonzept

Die Definition der vorgestellten Models macht es möglich, die abgelegten Inhalte explizit zu identifizieren. In diesem Zusammenhang wird der Aufbau der Struktur für Kategorie, Applikation und News im Folgenden näher erläutert. Die folgende Grafik zeigt die entwickelte Strukturierung in Verbindung mit verschiedenen Hierarchieebenen sowie mit Hilfe der angelegten Custom Content Models.

Abbildung 20 – Struktur der Inhalte



Quelle: Eigene Abbildung

Der Ursprung der hierarchischen Struktur beginnt bei der Wurzel. In diesem speziellen Fall ist dies der Abteilungsordner UH/I. Dieser verfügt über keinerlei spezifische Eigenschaften und ist somit vom Typ `cm:folder`. Wie bereits erwähnt sind die Kategorien von der Arbeitsgruppe Organisation und Kommunikation vorgegeben. Die gewählten Kategorien sind News und Applikation und sind in der zweiten hierarchischen Ebene angebracht. Es handelt sich um die speziellen Kategorie Ordner, deren Typ `kt:folder` aus dem oben beschriebenen Kategorie Model hervorgeht. Innerhalb des News Ordners befinden sich die einzelnen Nachrichten die wiederum auf eigene Ordner verteilt sind. Die Typen der Ordner für die Nachrichten sowie deren Ordnerinhalte sind im News Model beschrieben. Es handelt sich um `news:folder` beim Typ für den Ordner und um `news:document` sowie `news:image` für den Nachrichtentext und das dazugehörige Bild. Analog zu diesem Konstrukt sind die Ordner und Dokumente für die Applikationen strukturiert. Die Ordner sind vom Typ `appl:folder` und die entsprechenden Dokumente vom Typ `appl:document`.

Zusammenfassend bildet die entwickelte Struktur die Grundlage nach der die Inhalte zur internen Kommunikation im Content-Management-System abgelegt werden. Dabei ist zu beachten, dass begründet durch die manuelle Anlage des Contents über die Weboberfläche, der richtige Typ für die verschiedenen Arten auszuwählen ist. Bspw. muss ein Do-

kument, welches als Inhalt Neuigkeiten (News) enthält, entsprechend dem Typ `news:document` zugeordnet werden. Ist dies nicht der Fall und wird die vorgegebene Struktur verletzt, kann der Content nicht mehr eindeutig zugewiesen werden. Schließlich werden wie bereits erwähnt, die Hierarchie und die Metadaten verwendet, um dem Content Struktur zu verleihen. Das führt wiederum dazu, dass die automatische Verarbeitung durch die Java Applikation gestört ist, denn sie verwendet genau diese hierarchischen Informationen und die Metadaten zum Abruf der Inhalte. Wie genau der logische Zusammenhang und die daraus resultierende Darstellung aufgebaut ist, zeigt das folgende Kapitel.

## 6.2 Monitorapplikation L/information

Die Java Applikation wurde als Prototyp entwickelt und zeigt die Basistechnologien zum Abruf der Inhalte aus dem Content-Management-System. In diesem Zusammenhang wurde, wie bereits erwähnt, die openCMIS Client API verwendet, um das Java Programm an das Content Repository anzubinden. Aus diesem Grund wurde eine *Connector Klasse* implementiert, welche auf der einen Seite die Verbindung zum Content Repository aufbaut und auf der anderen Seite Methoden vorhält, um verschiedene Content Arten abzurufen. Bei den verschiedenen Content Arten handelt es sich um genau die Arten, die sich aus der vorgestellten Struktur ergeben. Darüber hinaus sind noch übergreifende Methoden definiert, welche wiederholt zur Abfrage eines Contents eingesetzt werden können. Ein Beispiel hierfür ist die Abfrage eines Bildes aus dem Alfresco System durch die Übergabe des Pfades, unter welchem dieses im Content Repository abgelegt ist. Der Abruf der Inhalte wird während des Programmverlaufes durchgeführt. Das bedeutet im Einzelnen, dass nicht der komplett im Repository gespeicherte Content beim Programmstart geladen wird, sondern nur genau dann, wenn dieser benötigt wird. Wie die Methoden zum Abrufen des Contents genau aufgebaut sind wird im Folgenden erläutert. Anschließend wird auf ausgewählte weitere Klassen eingegangen, die diesen Content in der Java Applikation L/information verarbeiten und darstellen. Dazu werden für die nächsten Kapitel grundlegende Kenntnisse der Java Programmierung vorausgesetzt.

## 6.2.1 Schnittstelle zu Alfresco – Klasse Connector

### 6.2.1.1 Verbindungsaufbau zum Content Repository

openCMIS ist eine Sammlung verschiedener Java Bibliotheken, Frameworks und Werkzeugen rund um den CMIS Standard. Folglich ermöglicht openCMIS, die im Kapitel 4.2.3 vorgestellten Content-Management Interoperability Services für Java Entwicklungen, sowohl serverseitig als auch clientseitig, zugänglich zu machen. Für die mit L/formationen verbundenen Ziele ist der Fokus auf die Client-Anbindung gelegt, da es sich bei dieser Applikation um eine Client Anwendung handelt. In diesem Zusammenhang stellt openCMIS zwei verschiedene Programmierschnittstellen, so genannte APIs (engl. application programming interface), zur Verfügung – die *Client API* und die *Client Bindings API*. Die Client API ist eine high-level, objektorientierte Schnittstelle und für die Mehrzahl der Anwendungen geeignet.<sup>60</sup> Die Client API wurde zur Anbindung von L/information an das Alfresco Content Repository verwendet. Die folgende Abbildung zeigt zugehörigen Quelltext, welcher die Verbindung zum Content Repository herstellt.

Abbildung 21 – Verbindungsaufbau; Auszug aus Connector.java

```
private static Session s;
private static String repID;

private Connector(){

    // default factory implementation of client runtime
    SessionFactory f = SessionFactoryImpl.newInstance();
    Map<String, String> parameter = new HashMap<String, String>();

    // user credentials
    parameter.put(SessionParameter.USER, user);
    parameter.put(SessionParameter.PASSWORD, password);

    // connection settings
    parameter.put(SessionParameter.ATOMPUB_URL, url);
    parameter.put(SessionParameter.BINDING_TYPE, BindingType.ATOMPUB.value());
    parameter.put(SessionParameter.REPOSITORY_ID, "96556066-fc45-4625-bc95-
b4ac27bd9ad0");

    // create session
    s = f.createSession(parameter);

    repoId = s.getRepositoryInfo().getId();
    System.out.println(s.getRepositoryInfo().getId());
}
```

Bereits beim Konstruktor-Aufruf der Klasse Connector wird die Verbindung zum Alfresco Content Repository hergestellt. Hierfür wird als erstes eine Session mit Hilfe der *SessionFactory* aufgebaut. Dieses Interface ist der Einstiegspunkt für die Client API und ist für die Erzeugung eines Session Objekts zuständig. Zu diesem Zweck ist es erforderlich, der

---

<sup>60</sup> Apache Chemistry 2011

Session verschiedene Parameter als Map zu übergeben. Die Parameter, welche zur erfolgreichen Verbindungsherstellung verwendet werden, sind Nutzernamen und Passwort, das Verbindungsprotokoll<sup>61</sup> sowie die RepositoryID. Anschließend kann die SessionFactory eine neue Session unter Zuhilfenahme der Parameter erzeugen. Nachdem die Verbindung erfolgreich hergestellt wurde, wird die RepositoryID abgefragt und zur Kontrolle auf der Konsole ausgegeben. Die erzeugte Session wurde für die Connector Klasse als privates Objekt definiert und ist folglich innerhalb der Klasse für alle Methoden verfügbar.

### 6.2.1.2 Abruf der Kategorien

Die ersten Inhalte, welche bereits nach dem Programmstart der Java Applikation aus dem Content-Management-System abgerufen werden, sind die Kategorien. Der folgende Quelltextauszug zeigt diese Methode und wird im Anschluss der Abbildung erläutert.

Abbildung 22 – Abruf der Kategorien; Auszug aus Connector.java

```
public List<Category> getCategories(){  
    ItemIterable<QueryResult> results = s.query("SELECT * FROM kt:folder", false);  
    List<Category> contAR = new ArrayList<Category>();  
    for(QueryResult result : results) {  
        Category cat = new Category();  
        cat.setTitle(result.getPropertyValueById(PropertyIds.NAME).toString());  
        cat.setObjectId(result.getPropertyValueById(PropertyIds.OBJECT_ID).toString());  
        cat.setObjectId(result.getPropertyValueById("kt:subtitle").toString());  
        contAR.add(cat);  
    }  
    return contAR;  
}
```

Wie bereits beschrieben, ist es durch die verschiedenen CMIS Dienste möglich, den Content durch eine SQL-Anweisung abzufragen. In diesem Kontext steht die erste Anweisung, welche mit der erzeugten Session eine Abfrage startet, die ein iterierbares Result-Set liefert. Die Abfrage `s.query("SELECT * FROM kt:folder", false)` liefert folglich alle Objekte vom Typ `kt:folder`. Im Anschluss wird in einer Schleife für jedes Objekt der SQL-Abfrage ein spezielles Kategorie Objekt angelegt. Die Klasse `Category` ist das Java Gegenstück zum Custom Content Model für die Kategorien. Folglich hält sie ebenso Attribute für einen Namen und eine Beschreibung, welche durch setter Methoden zugewiesen

---

<sup>61</sup> openCMIS verwendet das AtomPub Protokoll

werden. Zusätzlich wird die ObjectID jeder Kategorie aus dem Content Repository in der Klasse Category festgehalten. In der Abbildung ist zu sehen, dass die spezifischen Eigenschaften eines Kategorie Ordners, wie dieser im Custom Content Model beschrieben ist, mithilfe der dort definierten Typenbezeichnung abgerufen werden. Beispiel hierfür ist die Anweisung `result.getPropertyValueById("kt:subtitle")`. Jede dieser erzeugten Kategorie Klasse wird in einer Liste abgelegt, welche von der Methode zurückgegeben wird. Zusammenfassend erstellt die Methode eine Liste mit den einzelnen Kategorien und deren individuellen Eigenschaften. Die eigentliche Darstellung dieser ist reine Java Verarbeitung, auf welche im späteren Verlauf eingegangen wird.

### 6.2.1.3 Abruf der Applikationen

Alle in Alfresco angelegten Kategorien sind nach der vorgestellten Struktur innerhalb des Kategorie Ordners „Applikation“ abgelegt und vom Typ „appl:folder“. Wie oben beschrieben wird die ObjectID der Kategorie Ordner zwischengespeichert. Aus diesem Grund ist der Zugriff auf den Kategorie Ordner Applikationen mittels der ObjectID realisiert, wie der Quelltext der folgenden Methode zeigt.

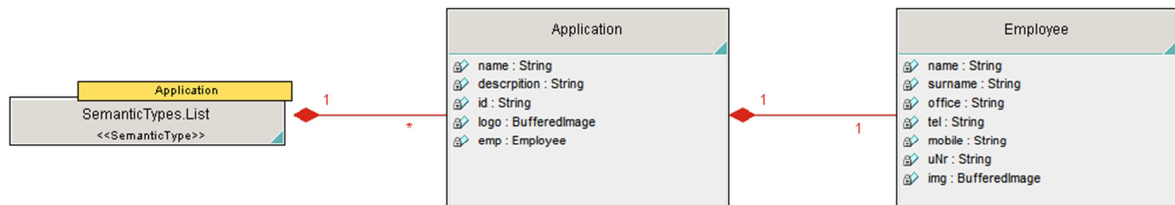
Abbildung 23 – Abruf der Applikation; Auszug aus Connector.java

```
public List<Application> getApplications(String objID) {  
  
    Folder folder = (Folder)s.getObject(s.createObjectId(objID));  
    ItemIterable<CmisObject> children = folder.getChildren();  
  
    List<Application> applications = new ArrayList<Application>();  
  
    for(CmisObject child : children){  
  
        Folder appl = (Folder)child;  
  
        CmisObject beschr = s.getObjectByPath(appl.getPath() +  
            "/Beschreibung.txt");  
        CmisObject logo = s.getObjectByPath(appl.getPath() + "/Logo.jpg");  
  
        Employee e = loadEmployee(child.getPropertyValue("appl:emp").toString());  
        Application app = new Application(child.getPropertyValue("appl:  
:appname").toString(), this.readTXT(beschr.getId()), child.getId(),  
            this.loadImage(logo.getId()), e);  
  
        applications.add(app);  
    }  
  
    return applications;  
}
```

Der Abruf der Applikationen aus dem Datenspeicher des Content Repositories von Alfresco ist ein wenig komplexer, da hier verschiedene Methoden verwendet werden um die Applikation in Java abzubilden. Aus diesem Grund ist es erforderlich, die vorliegende Methode Schrittweise zu erläutern. Auch hier werden zusätzliche Java Klassen implemen-

tiert, um die realen Objekte Applikation und Mitarbeiter zu implementieren. Ein Mitarbeiter Objekt (Klasse Employee) wird beim Abrufen einer jeden Applikation erstellt, weil der Applikationsverantwortliche wie bereits erwähnt, fest mit dieser verbunden ist. Um diese komplexen Zusammenhänge besser zu verdeutlichen, werden sie zunächst mit folgendem UML-Diagramm beschrieben.

Abbildung 24 – UML-Diagramm Applikationen



Quelle: Eigene Abbildung – erzeugt mit objectiF 7.1

Analog der bereits vorgestellten Methode zum Abruf der Kategorien, wird auch hier eine Liste erzeugt, welche mindestens eine oder unendlich viele Applikationen enthält. Im vorliegenden Fall handelt es sich um eine Komposition (Klasse A ist Teil von Klasse B). Die definierten Attribute einer Applikation sind wieder in Verbindung mit dem dazugehörigen Content Model zu betrachten. Dem hinzufügend hält die Klasse Application die ContentID aus dem Alfresco System und Attribute für das Applikationslogo und den Applikationsverantwortlichen. Folglich ist eine weitere Komposition definiert, in der genau ein Mitarbeiter ein Teil von genau einer Applikation ist. Mit dieser Vorbetrachtung ist es im Folgenden möglich, den in Abbildung 23 dargestellten Quelltext zu erläutern. Wie bereits beschrieben, ist die ObjectID des Ordners in dem die Applikationsordner abgelegt sind, bereits bekannt. Folglich erhält die Methode als Parameter diese ObjectID als Zeichenkette übergeben. Mit Hilfe dieser ID wird in der folgenden Anweisung der Zugriff auf den Ordner hergestellt. Anschließend wird mit der Methode `folder.getChildren()` der komplette Ordnerinhalt ausgelesen und einem iterierbaren Objekt zugewiesen. In diesem Zusammenhang entsteht demnach eine Art Liste, welche alle Applikationsordner enthält. Darauf folgend wird in einer Schleife jeder dieser Ordner ausgewertet. In dieser Schleife wird anfangs von der Session ein Objekt unter Zuhilfenahme des Pfades für die Applikationsbeschreibung und für das Applikationslogo erzeugt. Im nächsten Schritt wird der Mitarbeiter mithilfe der Methode `loadEmployee(child.getPropertyValue("appl:emp").toString())` erzeugt. Als Übergabeparameter wird die Personalnummer des Mitarbeiters verwendet, welche in den Metadaten der Applikationsordners hinterlegt wurde. Der Zugriff auf diese erfolge gemäß Content Model mittels `"appl:emp"`. Im Anschluss daran wird das Applikationsobjekt angelegt indem dem Konstruktor alle definierten Attribute übergeben werden. Hierbei besteht die Besonderheit, dass die Beschreibung und das



Logo durch die Methoden `this.readTXT(beschr.getId())` und `this.leseImage(logo.getId())` aus dem Content-Management-System abgerufen werden. Als Übergabeparameter werden die ObjectIDs der am Anfang der Schleife angelegten Objekte verwendet. Abschließend wird die erzeugte Applikation in der Liste abgelegt und nach dem Ende der Iteration wird diese Liste zurückgegeben.

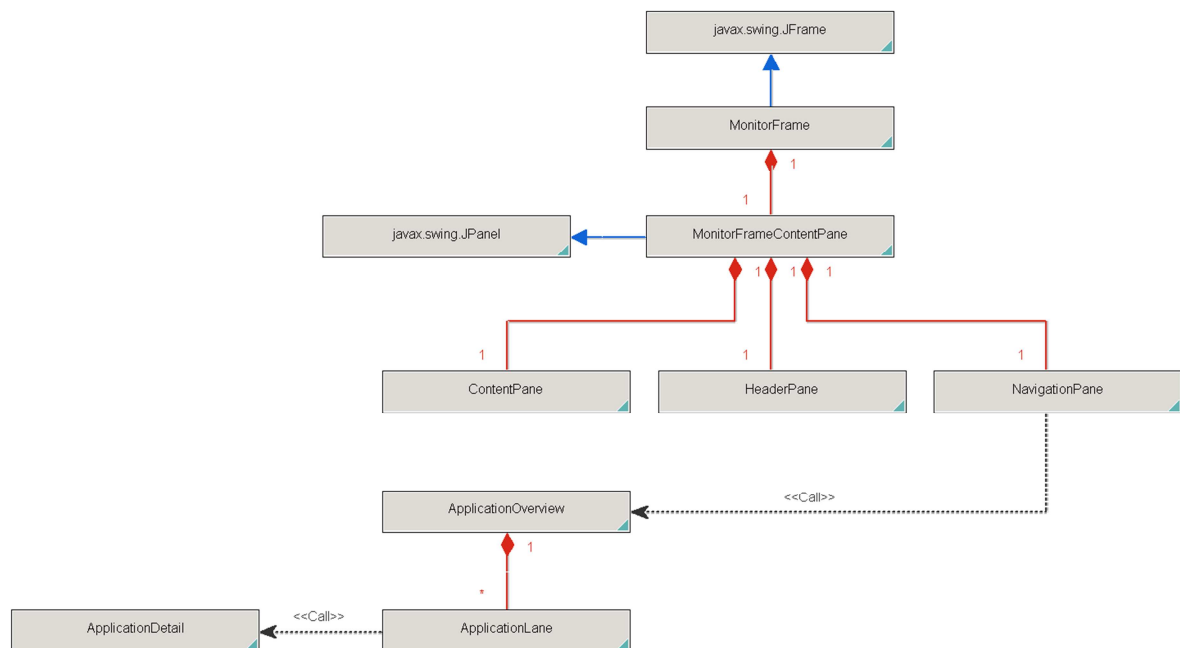
Wie genau die erzeugten Listen, welche die Kategorien oder die Applikationen enthalten verwendet werden, um deren Inhalte darzustellen, wird im folgenden Kapitel beschrieben.

## **6.2.2 Aufbau und Darstellung der Inhalte**

### **6.2.2.1 Fensteraufbau**

Die im Folgenden vorgestellten Klassen sind insbesondere mit der grafischen Oberfläche der Applikation verbunden. In diesem Kontext wird der Focus vor allem auf das Zusammenwirken der einzelnen Komponenten mit den abgerufenen Inhalten gelegt. Die grafische Oberfläche ist in verschiedene einzelne Bereiche geteilt, welche zur Laufzeit der Applikation entsprechend angepasst werden. Das folgende UML-Schema zeigt die verschiedenen Komponenten sowie deren Zusammenwirken zur Darstellung der Inhalte. Zu beachten ist, dass wie bereits im Kapitel 5.2.3 beschrieben, die Klassen nach dem Model-View-Control Prinzip strukturiert sind. Das bedeutet im Einzelnen, dass für jede in Abbildung 25 dargestellte Klasse, ein View, ein Model und ein Controller existiert. Zur Reduzierung der Komplexität wurde dieses Prinzip für die folgenden Betrachtungen vernachlässigt und es wird ausschließlich auf die View-Klassen eingegangen.

Abbildung 25 – Aufbau der Fensterstruktur



Quelle: Eigene Abbildung – erzeugt mit objectiF 7.1

Ausgangspunkt aller Grafikkomponenten des Hauptfensters ist die Klasse *MonitorFrame* welche von der Klasse *JFrame* abgeleitet wurde und somit die Basis eines Fensters bildet. Integriert in dieses Fenster ist ein *MonitorFrameContentPane*, bei dem es sich um ein Java *JPanel* handelt. Dieses Panel ist eine Containerklasse, in der verschiedene grafische Objekte abgelegt werden können. Diese Klasse ist dabei fest mit dem Hauptfenster verbunden, was aus der Komposition in der Abbildung hervorgeht. Das bedeutet, dass dem *MonitorFrame* genau ein *MonitorFrameContentPane* zugewiesen ist. Im *MonitorFrameContentPane* sind wiederum drei *ContentPanes* eingegliedert, welche ebenso als Komposition mit dem *MonitorFrameContentPane* in Beziehung stehen. Es handelt sich um genau ein *HeaderPane* ein *NavigationPane* und ein *ContentPane*. Diese Struktur erzeugt eine Dreiteilung des Containers in einen Kopfbereich, einen Navigationsbereich und einen Bereich für den Inhalt. Der Kopf dient zur Darstellung von Überschriften, einem Logo oder der Uhrzeit welche statisch durch die Applikation gesetzt werden. Im Navigationsbereich werden verschiedene Schaltknöpfe (engl. Buttons) dargestellt, um dem Anwender die Navigation zu ermöglichen. Im Contentbereich werden die eigentlichen Inhalte, welche in Verbindung mit den Kategorien stehen, angezeigt.

### 6.2.2.2 Darstellung der Kategorien

Wie bereits erwähnt, werden bereits beim Start der Anwendung die Kategorien aus dem Content-Management-System abgerufen. In diesem Zusammenhang sind bei der Erzeugung des Navigationsbereiches die Kategorien zu laden. Die folgende Methode beschreibt deren Verarbeitung sowie die Zuweisung zum NavigationPanel. Aufgrund der Anwendung des Model-View-Control Prinzips handelt es sich um den Konstruktor des *NavigationPaneModels*.

Abbildung 26 – Konstruktor Navigation Panel; Auszug aus *NavigationModel.java*

```
public NavigationModel() {  
  
    NavigationController c = new NavigationController(this);  
    view = new NavigationPane();  
  
    //Get categories  
    categories = Connector.getInstance().getCategories();  
  
    List<JButton> buttons = new ArrayList<JButton>();  
  
    for(Category cat : categories) {  
        JButton b = new JButton(cat.getTitle());  
        b.addActionListener(c);  
        b.setName(cat.getObjectId());  
        buttons.add(b);  
    }  
  
    //Add buttons  
    view.addButtons(buttons);  
}
```

Als erstes werden nach MVC im Konstruktor der Controller zur Interaktionsverarbeitung und der View erzeugt. Anschließend wird die vorgestellte Methode der Connector Klasse aufgerufen, welche die Liste der Kategorien liefert. Im nächsten Schritt wird eine leere Liste erzeugt, deren Inhalt Buttons sein sollen. In der folgenden Iterationsschleife wird für jede Kategorie ein Button erzeugt, dessen Titel die Kategorie ist. Als Name des Buttons wird die ObjektID verwendet. Aus diesem Grund, kann mit Hilfe des danach zugewiesenen ActionListeners der Button nicht nur durch den Titel, sondern auch mit der aus dem Content-Management-System stammenden ObjectID identifiziert werden. Abschließend wird der Button der Liste hinzugefügt und diese im View dargestellt.

Diese Konstruktion ermöglicht es, bei einem ActionEvent einer dieser Buttons, direkt die interne ObjektID der Kategorie Ordner des Alfresco Repositories zu verwenden. Dies wiederum wurde, wie bereits bei der vorgestellten Methode zum Abruf der Applikationen erwähnt, berücksichtigt. Die Darstellung der Applikationen mittels einer Übersicht sowie einer Detailansicht wird im folgenden Kapitel beschrieben.

### 6.2.2.3 Darstellung der Applikationen

Die Darstellung der Applikationen, sowie deren individuelle Eigenschaften werden wie im Klassendiagramm auf Seite 61 zu sehen, mittels drei verschiedenen Komponenten realisiert. Es handelt sich um den *ApplicationOverview*, welcher wiederum aus mehreren *ApplicationLanes* besteht, und den *ApplicationDetail*. Die Applikationsübersicht, die mit der *ApplicationOverview* Klasse implementiert ist, wird durch das vorgestellte *NavigationPanel* aufgerufen. Der dadurch angestoßene Konstruktor erzeugt analog zum vorgestellten *NavigationPanel* einen Controller und einen View. Teil dieser Klasse sind so genannte Applikationsstreifen, welche eine Kurzdarstellung einer Applikation realisieren. Ferner wird im Konstruktor dem *ContentPane* eine Komponente zur Darstellung zugewiesen. Der Zusammenhang zwischen Übersicht mit einzelnen Applikationsstreifen und die Applikationsdetailansicht soll durch den folgenden Quelltext näher erläutert werden.

Abbildung 27 – Darstellung der Applikationen; Auszug aus *ApplicationOverviewModel.java*

```
public void loadApplications(String objID){  
  
    Connector c = Connector.getInstance();  
    applications = c.getApplications(objID);  
  
    view.getContent().removeAll();  
  
    for(Application app : applications){  
  
        ApplicationLane aLane = new ApplicationLane(app.getLogo(),  
            app.getAppName(), app.getDescription(),app.getId());  
        aLane.addMouseListener(controller);  
  
        view.addApplicationLane(aLane);  
  
    }  
  
}
```

Zum Abruf und zur Darstellung der Applikation wird die *ObjectID* des Kategorie Ordners verwendet, in dem diese sich befinden. Anschließend werden mittels der im *Connector* verfügbaren Methode die Applikationen abgerufen. Danach wird in einer Schleife für jede Applikation eine *ApplicationLane* angelegt. Eine *ApplicationLane* ist Panel, welches sich wie ein Streifen vorzustellen ist, der das Applikationslogo, den Applikationsnamen und die Beschreibung zeigt. Zusätzlich zu den genannten, wird die Applikations-ID aus dem Content Repository gespeichert. Diesen einzelnen Streifen wird ein Controller hinzugefügt und sie werden anschließend im *ApplicationOverview* untereinander dargestellt. Der Controller überwacht für jede einzelne *ApplicationLane* die Mauseingaben und ist durch die mit der Lane verbundenen Applikations-ID in der Lage, diese zu identifizieren. In diesem Zusammenhang wird die Methode `showAppDetail(String objID)` aufgerufen und die Detailseite der Applikation angezeigt, wie es folgender Quelltext belegt.

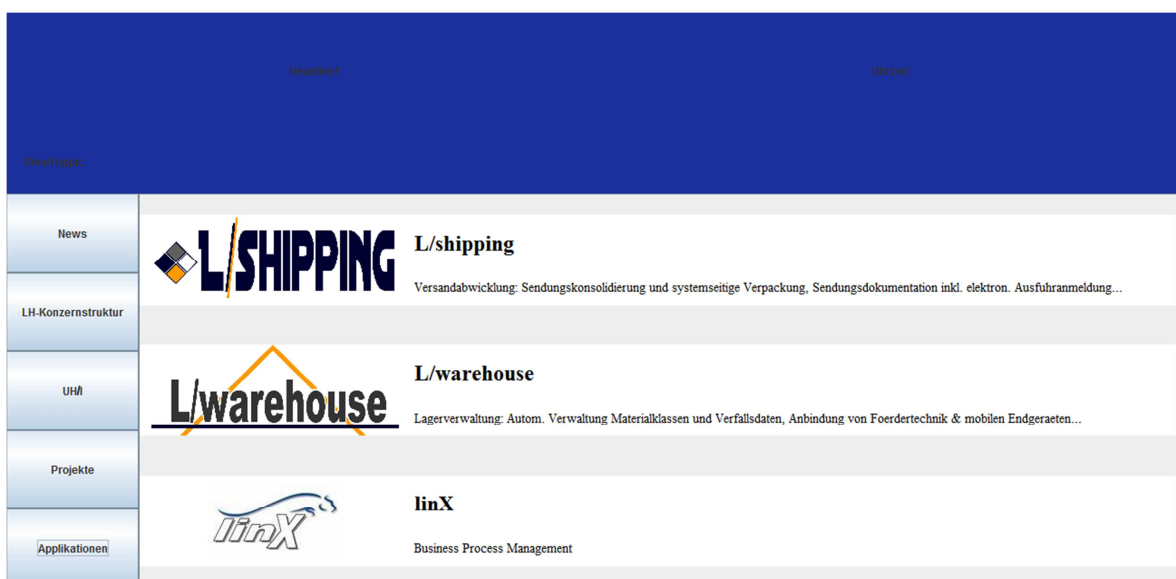
Abbildung 28 – Detailansicht einer Applikation

```
public void showAppDetail(String objID){
    Application selectetApp = null;
    for(Application app : applications){
        if(app.getId().equals(objID)){
            selectetApp = app;
        }
    }
    if(selectetApp != null){
        ModelRegistry.get(ApplicationDetailModel.class).setApplicationDetails (selectetApp);
        ModelRegistry.get(ContentModel.class).showComponent(MainContentNames.APPDETAIL);
    }
}
```

Wie in Abbildung 28 zu erkennen wird ein temporäres Objekt definiert, welche die ausgewählte Applikation widerspiegeln soll. Anschließend wird mit Hilfe der ID die konkrete Applikation in der Applikationsliste gesucht und dem temporären Objekt zugewiesen. Folglich wird dem ApplikationDetail die bestimmte Applikation übergeben und die Applikationsdetails werden im ContentPane angezeigt.

Die vorgestellte Darstellung der Inhalte aus dem Content-Management-System ist ein Zusammenwirken aus Metadaten, welche aus dem Content Repositoy stammen und den Java Datenstrukturen denen diese zugewiesen werden. Aufgrund der dadurch entstehenden Komplexität, verbunden mit der Vielzahl an Fensterkomponenten, soll die folgende Grafik den Aufbau erneut verdeutlichen.

Abbildung 29 – L/information; Applikationsübersicht



## 6.3 Weiterentwicklung und Ausblick

Für den in den vorangegangenen Kapiteln vorgestellten Prototypen, welcher die Machbarkeit des Softwarevorhabens demonstrieren soll, ergeben sich folgende Schritte um diesen weiter zu entwickeln. Die komplette Migration der Inhalte auf Basis der bestehenden PowerPoint Präsentation in das Alfresco Content-Management-System ist als erster Schritt zu sehen. Dazu sind die Inhalte in die beschriebene Content Struktur einzugliedern. Nach der kompletten Überführung und der damit verbundenen Definition weiterer Content Models kann die Java Applikation entsprechend erweitert werden. In diesem Zusammenhang ist die grafische Oberfläche den Corporate Design Vorgaben der Lufthansa Technik Logistik GmbH anzupassen. Ferner sind für die neu migrierten Inhalte ebenso ContentPanels zu definieren, welche die Inhalte auf geeignete Weise darstellen.

Das im Zusammenhang mit dieser Arbeit entwickelte IT-Konzept birgt für die zukünftige Anwendung erhebliches Potential. Das Softwaresystem könnte abteilungsübergreifend eingesetzt werden, indem die Java Applikation nicht wie in diesem Fall das Wurzelverzeichnis UH/I, sondern den einer anderen Abteilung anwendet. In diesem Zusammenhang kann das Alfresco System für verschiedene Mitarbeiter zur Pflege der Informationen entsprechend angepasst werden. Daraus folgt die Verwendung des Alfresco Berechtigungssystems, welches es ermöglicht, verschiedenen Nutzern den Zugriff auf verschiedene Ordner zu gewähren. Somit besteht die Möglichkeit, dass bspw. der Applikationsverantwortliche einer Applikation die Informationen selbst pflegt und in Verbindung mit dem Berechtigungssystem nur den Zugriff auf „seinen“ Applikationsordner erhält. Des Weiteren ist damit die Ablage des Contents in einen falschen Ordner, oder gar das Löschen von Informationen anderer Kategorien für den Mitarbeiter unterbunden. Für die Zukunft wäre weiterhin denkbar, dass begründet durch die Anwendung der CMIS Standards, Informationen direkt aus den Operativsystemen oder Reportingsystemen der LTL in das Content Repository von Alfresco zu laden. Unter diesen Umständen sind Fehlverhalten der Applikation verbunden mit der manuellen Pflege ausgeschlossen. Zudem wären die Informationen, gerade in Bezug auf das Reporting hoch aktuell. Eine Migration der Applikation L/information in das Intranet der Lufthansa Technik Logistik GmbH würde den Anwenderkreis erweitern und ist durch die Anwendung von Java denkbar.

## 7. Zusammenfassung

Das entwickelte und vorgestellte IT-Konzept bildet die Basis für die erfolgreiche interne Kommunikation bei der Lufthansa Technik Logistik GmbH. Es wurde eine Möglichkeit zur einfachen, effektiven und vor allem nachhaltigen Datenpflege gefunden und deren Umsetzbarkeit bereits in einem Prototypen vorgestellt. Die Architektur des Softwarevorhabens ermöglicht die Speicherung der Informationen und die Darstellung derer, getrennt voneinander zu steuern. Die vielseitigen Möglichkeiten des gewählten Content-Management-Systems bieten zudem noch erhebliches Potential, das Konzept auf andere Bereiche auszudehnen oder gar standortübergreifend einzusetzen. Durch die Anwendung einer Java Applikation kann diese den individuellen Anforderungen, sowohl aus technischer, als auch aus organisatorischer Sicht angepasst werden.

Es wurde bereits darauf eingegangen, dass in manchen Unternehmen die interne Kommunikation vernachlässigt wird. Das mit dieser Arbeit entwickelte Konzept ermöglicht es, gut und effizient intern zu kommunizieren und das in einer Art und Weise in der viele notwendige Schritte automatisiert ablaufen. Verbunden mit den im Ausblick gegebenen Möglichkeiten der automatisierten Datenpflege durch die Anbindung bereits bestehender Systeme, ist in bestimmten Sektionen die Chance interne Kommunikation vollautomatisch zu bewerkstelligen gegeben.

Die interne Kommunikation, bei der die Informationen in Interaktion mit dem Anwender abgerufen werden können, ist zudem ein erheblicher Vorteil. Der Anwender ist in der Lage, die Informationen zu genau dem Zeitpunkt abzurufen, wenn sie für ihn interessant sind. Diese personalisierte Art der internen Kommunikation verbunden mit den vorgestellten Einsatz- und Erweiterungsmöglichkeiten bilden eine optimale Grundlage ein Softwaresystem zur internen Kommunikation einzuführen.

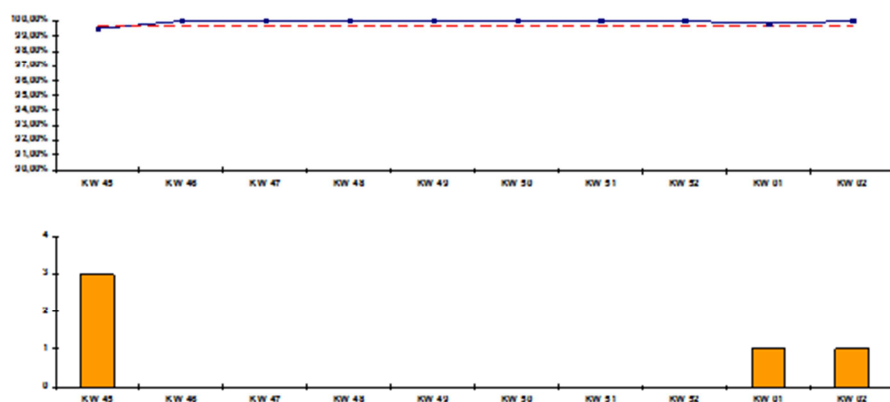
# Anhang

Auf den folgenden Seiten sind die PowerPoint Folien der derzeit auf dem UH/I Monitor laufenden Präsentation dargestellt.

## Systemverfügbarkeiten

### Systemverfügbarkeit L/warehouse

Verfügbarkeit





L/warehouse

Geplante Auszeiten



19.01.2011	12:30 – 13:30 Uhr	Release
06.02.2011	00:00 – 02:00 Uhr	SAP Kernel Patch

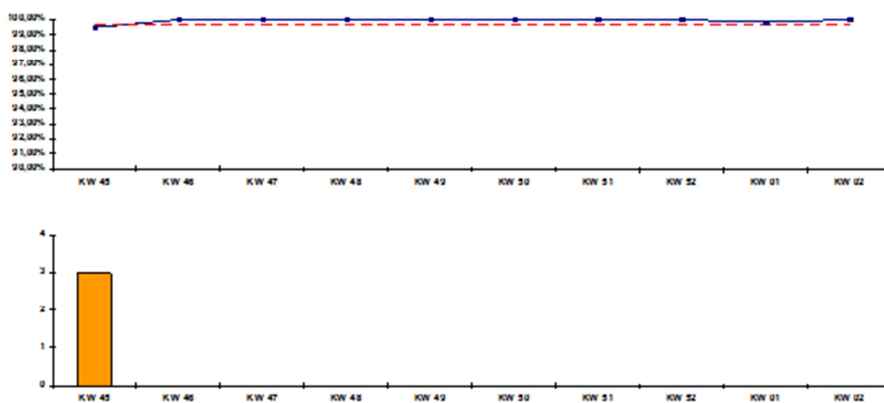
## Wussten Sie eigentlich, dass...



L/warehouse 1.007.554 Stammdaten verwaltet?

Systemverfügbarkeit L/shipping

Verfügbarkeit



Geplante Auszeiten



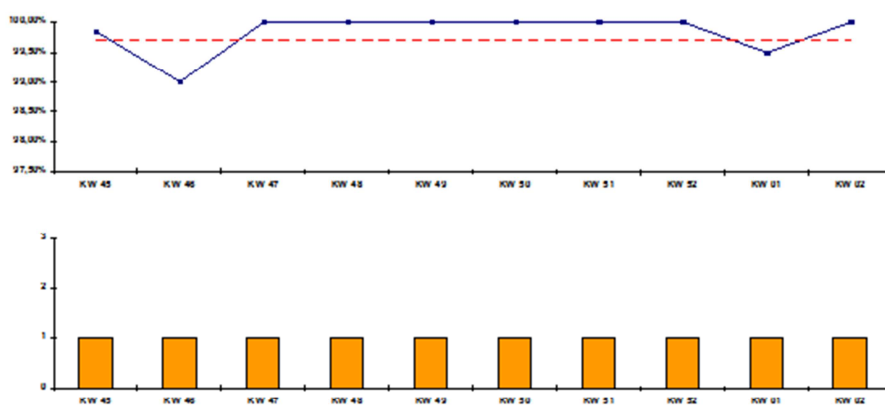
19.01.2011	12:30 – 13:30 Uhr	Release
06.02.2011	00:00 – 02:00 Uhr	SAP Kernel Patch

## Wussten Sie eigentlich, dass...



die durchschnittliche Verfügbarkeit von L/shipping im Jahr 2009 98,97% betrug?

Verfügbarkeit



Geplante Auszeiten



25.01.2011

05:00 – 06:00 Uhr

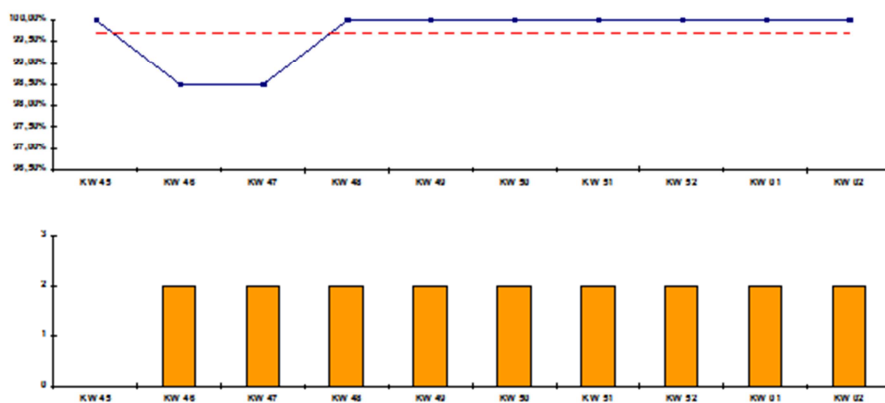
Release

**Wussten Sie eigentlich, dass...**



*linX ca. 110.000€ Betriebskosten im Jahr verursacht?*

Verfügbarkeit



TELOS

Geplante Auszeiten



11.01.2011

11:45 – 12:15 Uhr

Bugfix

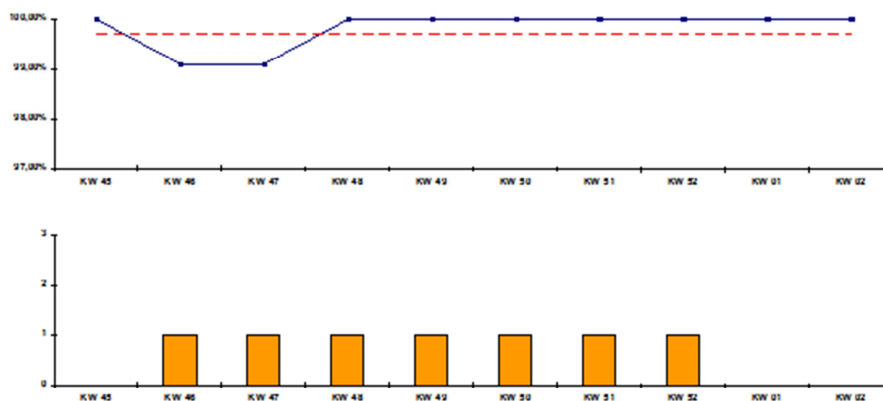
## Wussten Sie eigentlich, dass...



TELOS 11 fachliche Schnittstellen hat?

Systemverfügbarkeit B2Log

Verfügbarkeit



Geplante Auszeiten



06.12.2010 – 02.01.2011

Frozen Zone

## Wussten Sie eigentlich, dass...



*B2Log 2009 304.920 Frachtbriefe erstellt hat?*



19.01.2011

(12:30 – 13:30 Uhr) L/warehouse, L/shipping  
Release

25.01.2011

(05:00 – 06:00 Uhr) LinX  
Release

06.02.2011

(00:00 – 02:00 Uhr) L/warehouse, L/shipping  
SAP Kernel Patch

Wussten Sie eigentlich, dass...

*tagPilot im Jahr 2009  
durchschnittlich zu 99,72%  
verfügbar war ?*



**Blick über den Tellerrand**

### Biokraftstoff im Flugbetrieb

29.11.10



Lufthansa wird als weltweit erste Airline Biokraftstoff im normalen Flugbetrieb einsetzen. Ab April wird ein Airbus A321 für sechs Monate auf der Strecke Hamburg-Frankfurt-Hamburg mit bio-synthetischem Kerosin fliegen. Dabei wird eines der zwei A321-Triebwerke von einer Treibstoffmischung mit einem Anteil von 50 Prozent bio-synthetischem Kerosin gespeist werden.

LHT/red

### EasyJet bestellt 15 neue Airbus A320

06.01.11



Die britische Lowcost-Airline Easyjet hat eine Option auf 15 Airbus-Flugzeuge vom Typ A320 in eine feste Bestellung umgewandelt. Die Flotte des Unternehmens soll von gegenwärtig 196 Flugzeugen bis Ende 2013 auf 220 Maschinen wachsen. EasyJet wird dann ausschließlich Fluggeräte der A320-Serie auf ihren Strecken einsetzen. Durch den Einsatz größerer Flugzeuge könne der Billigflieger die Stückkosten pro Sitz senken.

LHT/red

#### Lufthansa stellt ein: 4.000 neue Mitarbeiter in Deutschland

06.01.11



Lufthansa wird 2011 rund 4.000 neue Mitarbeiter in Deutschland einstellen. 2.200

Flugbegleiter/innen, 900 neue Servicemitarbeiter und 270 Pilotenschüler.

Mit dem Airbus A380 wächst der Personalbedarf im Cockpit, in der Kabine und am Boden. Lufthansa zählt zu den attraktivsten Arbeitgebern Deutschlands, rund 115.000 Bewerbungen wurden im vergangenen Jahr über das Karriereportal abgegeben.

LHT/red

#### 100 Jahre Hamburg Airport – Rekordzahlen 2010

10.01.11



Am 10. Januar 1911 wurden im Dorf Fuhlsbüttel die Hamburger Luftschiffhallen für den Bau eines Landeplatzes gegründet. Dies war die Geburtsstunde des Hamburger Flughafens.

Im Jahr 2010 nutzten 12,96 Millionen Passagiere den Flughafen. Das sind rund 6 Prozent mehr als im Jahr 2009. Im Jahr 2010 boten 60 Fluggesellschaften ab Hamburg direkte Flüge zu 115 Zielen an. Die Starts und Landungen lagen bei durchschnittlich 430 pro Tag.

LHT/red



**Information Technology**  
HAM UH/I

**Dean**  
Dr. Andreas Roth - 5304

**Information Management**  
Head of Department: Dr. Roth - 5304  
Staff: Dr. Roth - 5304

**Information Technology**  
Head of Department: Dr. Roth - 5304  
Staff: Dr. Roth - 5304

**Computing & Technology**  
Head of Department: Dr. Roth - 5304  
Staff: Dr. Roth - 5304

**Information Management**  
Head of Department: Dr. Roth - 5304  
Staff: Dr. Roth - 5304

**Information Technology**  
Head of Department: Dr. Roth - 5304  
Staff: Dr. Roth - 5304

**Computing & Technology**  
Head of Department: Dr. Roth - 5304  
Staff: Dr. Roth - 5304

Herzlich willkommen bei UH/I



Die Aufgabe der Abteilung HAM UH/I ist die durchgehende Bereitstellung speziell auf die MRO-Logistik ausgerichteter IT-Applikationen. Internationale Logistikprozesse werden mit einzigartigen Funktionen optimal unterstützt. Die volle Steuerungsfähigkeit entlang des Logistikprozesses ist unser Ziel. Ein aussagekräftiges Monitoring wird bereitgestellt. HAM UH/I entwickelt diese Fähigkeiten für die MRO-Logistik in zahlreichen Projekten entsprechend den Bedürfnissen der Kunden weiter.

Die Informationstechnologie der LTL sind aber nicht nur IT-Applikationen, -Server oder -Projekte, sondern diese Informationstechnologie wird durch die Mitarbeiterinnen und Mitarbeiter der Abteilung repräsentiert, die sich Ihnen mit ausgewählten nützlichen oder interessanten Informationen auf den folgenden Folien vorstellen möchten.

Ihr Dr. Andreas Roth (Leiter, UH/I)

Unsere Gruppen

**ILOG** (Informationslogistik)  
Gruppenleiterin: **Friederike Hoffmann**

Das Team ILOG kümmert sich um IT-Applikationen im Bereich Informationslogistik. Dies sind Applikationen, die den physischen Logistikprozess und dessen Steuerung entweder mit zusätzlichen Informationen versorgen und unterstützen oder aber Datennutzer derselben sind, um Informationen zu Reporting-, Monitoring- und kaufmännischen Zwecken weiterzuverarbeiten.



Hierzu zählen das SAP-System der LTL, der LTLs und der LTLA, das Dokumentenmanagementsystem, verschiedene Tracking- und Tracing-relevante Applikationen, die RFID-Middleware, das Archivierungssystem, die zentrale Datendrehscheibe, das Abrechnungssystem und Business-Intelligence-Anwendungen.

#### Unsere Gruppen

##### **PLOG** (Physische Logistik)

Gruppenleiter: **Dr. Andreas Schirmer**

Wir verantworten die Applikationen der LTL rund um die physischen Logistikprozesse. Dazu gehören die Prozesse in den Bereichen Wareneingang, Warehouse Management und Versand, die Zollverfahrensermittlung, -bestandsführung und -kommunikation sowie die Dokumentation und Abrechnung externer Transporte. In unserer Verantwortung liegen sowohl der stabile, sichere Betrieb als auch die nachhaltige fachliche

Entwicklung unserer Applikationen; die Aufgaben der Applikationsverantwortlichen und Projektleiter reichen dabei von Konzeption und Realisierung über Tests und Abnahme bis zu Produktivsetzung, Rollout und Optimierung. In allen Bereichen nutzen wir auch die Erfahrung interner und externer Partner.

Im Rahmen weltweiter Logistikprojekte der LTL für konzerninterne und -externe Kunden koordinieren unsere Projektleiter im Team Customer Solutions die applikationsübergreifende Aufnahme, Abstimmung und Umsetzung aller IT-bezogener Anforderungen.



#### Unsere Gruppen

##### **SLOG** (Steuerungslogistik)

Gruppenleiterin: **Dr. Stephanie Nagel-Heyer**

Die Gruppe SLOG beschäftigt sich mit drei Aufgabenpaketen: Im Rahmen der Planung und Durchführung von IT-Projekten, die sich mit der Steuerung und zeitlichen / prozessualen Planung von innerbetrieblichen Logistikprozessen sowie internen und externen Transporten beschäftigen, stehen wir unseren internen wie externen Kunden sowohl im nationalen als auch im internationalen Bereich als Berater und Projektmanager zur Seite.



Die Applikationsverantwortlichen sind für den technischen Betrieb des Business-Process-Management-Systems linX verantwortlich und gewährleisten zusammen mit dem Helpdesk, dem SCD und dem externen Dienstleister SALT Solutions GmbH einen zeitnahen Support im Störfall.

Über die Mitarbeit in der Betreiberorganisation linX arbeitet die Gruppe aktiv an der Analyse und Behebung bestehender Problemfälle sowie an der kontinuierlichen Weiterentwicklung des Systems linX mit.

#### Unsere Leitlinien



Wir bauen Brücken zwischen fachlichen Anforderungen und technischen Lösungen.

Unsere Mitarbeiter sind die Basis unseres Erfolgs und werden gezielt gefördert.

Durch innovative IT-Lösungen stärken wir die Marktposition der LTL.

Enge Teamarbeit, Flexibilität und ein konstruktiver Umgang mit Fehlern sind unsere Erfolgsfaktoren.

Kundenanforderungen werden unter Einsatz standardisierter Methoden umgesetzt.

Durch Transparenz und Dokumentation erzielen wir Know-how-Transfer.

#### Auswahl weiterer Aktivitäten

(Reihenfolge entspricht Priorität oder Umfang der Projekte, zudem alphabetisch)

- Abbildung Prozesse für Außenlager
- Absicherung Sendungsaufösungen (L/shipping)
- Absicherung Zollstatusänderungen im Versand
- Erarbeitung einer Lösung für die Terminierung von mehrpositionalen Aufträgen (linX)
- Erstellung NLV-übergreifendes internes Reporting in L/report
- Implementierung einer Lösung in linX zur Behebung des Prozessabbrisses in LOTS bei Vorgängen es Wareneingang für externe Kunden
- LFE - Einlagerung nach Verschrottungsbuchung P04
- Projektvorbereitung UH/C Ergebnisrechnung in L/report
- Reorganisation Altdaten L/report
- Reporting Malta
- Taskforce „Roter Sendungsmonitor im Versand HAM“
- Überarbeitung Aktivierungsregeln Lager FRA (L/warehouse)
- Vorbereitung ATLAS 8,3
- Vorbereitung Tracking & Tracing-Plattform für Direktkundentemplate
- Weiterentwicklung des Testverfahrens zum Einsatz von HP Quicktest bei linX



## Außerhalb UH/I

### Neu bei LTL

*Liebe Kolleginnen und Kollegen,*

seit dem 01. Dezember 2010 hat HAM UH/TN4 Verstärkung durch Frau Evelyn Suske erhalten. Frau Suske wird als Projektingenieurin Transport- und Netzmanagement Projekte für die LHT unterstützen und bei der Konzepterstellung für Direktkunden mitwirken.

Evelyn Suske studierte an der HAW Hamburg technische BWL mit den Schwerpunkten Materialwirtschaft/Logistik. Vor dem Studium absolvierte sie eine Ausbildung zur Industriekauffrau bei der Schleich GmbH und war anschließend in der zentralen Produktionsplanung tätig. Während des Studiums sammelte sie im Rahmen von Praktika bei der Daimler AG und Brose Ltd. im In- und Ausland Erfahrungen in der Logistikplanung.

Frau Suske ist erreichbar unter:

Telefon: 040 / 5070 – 65869  
E-Mail: [evelyn.suske@lth.dlh.de](mailto:evelyn.suske@lth.dlh.de)  
Gebäude/Raum: G256 / R454



***Wir wünschen Frau Suske viel Erfolg !***



**Neu bei LTL**

*Liebe Kolleginnen und Kollegen,*

seit dem 18. Oktober 2010 hat HAM UH/TS1 Verstärkung durch Herrn Nikola Karamalakov erhalten. Herr Karamalakov wird die Transportkoordination für das Projekt NEK (Neue Europäische Kabine) übernehmen. Herr Karamalakov studierte an der Technischen Hochschule Wildau Wirtschaftsingenieurwesen mit den Schwerpunkten Informatik / Operations Research und Supply Chain Management.

Im Sommer 2009 absolvierte er erst ein neunwöchiges Sommerpraktikum und anschließend ein sechsmonatiges Praktikum bei der Lufthansa Technik Sofia (LTSF).

Herr Karamalakov ist erreichbar unter:

Telefon: 040 / 5070 – 64711  
E-Mail: nikola-dimitrov.karamalakov@ltl.dlh.de  
Gebäude/Raum: G210 / R180



***Wir wünschen Herrn Karamalakov viel Erfolg !***

**Neu bei LTL**

*Liebe Kolleginnen und Kollegen,*

seit dem 01. Januar 2011 hat die Niederlassung Frankfurt Verstärkung durch Herrn Oliver Günther als Produktionsingenieur im Bereich Lager Service (FRA UE/BW) erhalten. Oliver Günther studierte Wirtschaftsingenieurwesen mit den Schwerpunkten Materialfluss und Logistik an der Fachhochschule Erfurt. Nach Abschluss seines Studiums im Jahr 2008 arbeitete Herr Günther als Logistikplaner bei der Benteler Automobiltechnik GmbH. Sein Aufgabenbereich umfasste u.a. die Vor- und Nachbereitung von weltweiten Frachtausschreibungen sowie die Koordination von Sonderfrachten. Zuletzt war er in der Funktion für die Planung des logistischen Flusses für Projektneuanläufe zuständig.

Herr Günther ist erreichbar unter:

Telefon: 069 / 696 – 90019  
E-Mail: oliver.guenther@ltl.dlh.de  
Gebäude/Raum: G401 / R214



***Wir wünschen Herrn Günther viel Erfolg !***

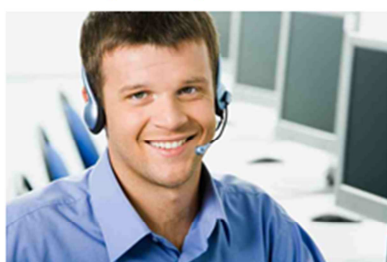


# Info



## Service Desk

Ihr Ansprechpartner für alle IT Probleme weltweit (24/7)



Sie erreichen uns unter:

<b>HAM</b>	<b>-5252</b>
<b>FRA</b>	<b>-4000</b>
<b>RoW</b>	<b>-67620</b>

# Literatur

Alfresco Software 2011	Alfresco Software, Ltd: <a href="http://www.alfresco.com/de/products/">http://www.alfresco.com/de/products/</a> , verfügbar am 12.01.2011, 15:00 Uhr
Apache Chemistry 2011	Apache Chemistry: <a href="http://incubator.apache.org/chemistry/opencmis.html">http://incubator.apache.org/chemistry/opencmis.html</a> , verfügbar am 19.01.2011
Balzert 2001	Balzert, Helmut: <i>Lehrbuch der Software-Technik</i> , Heidelberg, Spektrum Akad. Verl. 2001
Balzert 2009	Balzert, Helmut: <i>Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering</i> , Heidelberg, Spektrum Akademischer Verlag 2009
Caruana 2010	Caruana, David: <i>Professional alfresco</i> , Indianapolis, Wiley 2010
Dr. Roth – LTL 2010	Dr. Andreas Roth: Abteilungsleitung Informationstechnologie der Lufthansa Technik Logistik GmbH
Ernst, Ernst et al. 2009	Ernst, Nadin; Ernst, Nadin; Zerfaß, Ansgar; Möslin, Kathrin M.: <i>Kommunikation und Innovation in deutschen Unternehmen /// Kommunikation als Erfolgsfaktor im Innovationsmanagement</i> , Wiesbaden, Gabler Verlag / GWV Fachverlage GmbH Wiesbaden 2009
Gadatsch 2010	Gadatsch, Andreas: <i>Grundkurs Geschäftsprozess-Management</i> , Wiesbaden, Vieweg + Teubner 2010
Hasenkamp, Stahlknecht 2010	Hasenkamp, Ulrich; Stahlknecht, Peter: <i>Einführung in die Wirtschaftsinformatik</i> , Berlin [u.a.], Springer 2010
Hoffmann 2001	Hoffmann, Claus: <i>Das Intranet</i> , Konstanz, UVK 2001
Krcmar 2005	Krcmar, Helmut: <i>Informationsmanagement</i> , Berlin, Springer 2005
Krüger 2008	Krüger, Guido: <i>Handbuch der Java-Programmierung</i> // Guido Krüger., München, Boston, Addison-Wesley 2008
Künzel 2005	Künzel, Hansjörg: <i>Handbuch Kundenzufriedenheit</i> , Berlin, Springer 2005
LTL, 2010	Lufthansa Technik Logistik GmbH: Das Unternehmen1 2010, <a href="http://www.lufthansa.com/online/portal/lh_ltl/de/nonav/homepage?ctest=125562461176">http://www.lufthansa.com/online/portal/lh_ltl/de/nonav/homepage?ctest=125562461176</a> verfügbar am 18.12.2010, 15:00 Uhr
Musch 2010	Musch, Sebastian: <i>Praxissemesterbeleg: Aufzeigen von Einsparpotentialen bei der Einführung eines Transportmanagementsystems für den Zentralversand Hamburg</i> 2010
Prexl 2010	Prexl, Anja: <i>Nachhaltigkeit kommunizieren - nachhaltig kommunizieren</i>



- ren, Wiesbaden, VS Verlag für Sozialwissenschaften / GWV Fachverlage GmbH Wiesbaden 2010
- Puttenat 2007 Puttenat, Daniela: *Praxishandbuch Presse- und Öffentlichkeitsarbeit*, Wiesbaden, Betriebswirtschaftlicher Verlag Dr. Th. Gabler | GWV Fachverlage GmbH Wiesbaden 2007
- Riggert 2009 Riggert, Wolfgang: *ECM - Enterprise Content Management*, Wiesbaden, Vieweg+Teubner Verlag / GWV Fachverlage GmbH Wiesbaden 2009
- Schmitz 2007 Schmitz, Roland: *Kompendium Medieninformatik*, Berlin, Heidelberg, Springer 2007
- Schuster, Wilhelm 23.12.2000 Schuster, Erwin; Wilhelm, Stephan: *Content Management*,
- Soucek 2006 Soucek, Roman: *Eskalation von Ziel- versus Handlungscommitment*, Hamburg, Kovac 2006
- Stahlknecht, Hasenkamp 2005 Stahlknecht, Peter; Hasenkamp, Ulrich: *Einführung in die Wirtschaftsinformatik*, Berlin, Heidelberg, Springer Berlin Heidelberg 2005
- Stetina, Kryspin-Exner 2009 Stetina, Birgit U.; Kryspin-Exner, Ilse: *Gesundheit und Neue Medien*, Vienna, Springer-Verlag Vienna 2009
- Weder 2003 Weder, Jürgen: *Betriebliche Anwendungen der Informatik*, Nordstedt, Books on Demand GmbH 2003
- Wolf 2010 Wolf, Guido: *Der Business Discourse*, Wiesbaden, Gabler 2010

# Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, den 10. Februar 2011

Sebastian Musch